



André Filipe Lucas Ricardo

Licenciatura em Engenharia Informática

Instalação Interactiva para Exploração de Iluminuras Medievais

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Prof. Doutor Nuno Manuel Robalo Correia, Prof. Associado com Agregação, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

Júri:

Presidente: Doutor José Augusto Legatheaux Martins

Vogais: Doutor Manuel João Caneira Monteiro da Fonseca
Doutor Nuno Manuel Robalo Correia



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

SETEMBRO, 2011



André Filipe Lucas Ricardo

Licenciatura em Engenharia Informática

Instalação Interactiva para Exploração de Iluminuras Medievais

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Prof. Doutor Nuno Manuel Robalo Correia, Prof. Associado com Agregação, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

Júri:

Presidente: Doutor José Augusto Legatheaux Martins

Vogais: Doutor Manuel João Caneira Monteiro da Fonseca
Doutor Nuno Manuel Robalo Correia



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

SETEMBRO, 2011

Instalação Interactiva para Exploração de Iluminuras Medievais

Copyright © André Filipe Lucas Ricardo, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Em memória do meu avô José Lucas e da minha tia Silvina.

Até sempre.

Agradecimentos

Antes de mais gostaria de agradecer ao Prof. Doutor Nuno Correia pela oportunidade de participar neste projecto de investigação e pela orientação, disponibilidade e ajuda prestada, quer no desenvolvimento das diferentes aplicações incluídas nesta dissertação, quer na elaboração deste documento.

Gostaria de agradecer a todos os elementos que participaram neste projecto, em particular à Rita Carvalho, ao Tarquínio Mota, à Rita Castro, à Maria João Melo, à Catarina Miguel, à Maria Adelaide Miranda e à Inês Correia.

Gostaria de agradecer ao Centro de Informática e Tecnologias de Informação (CITI) da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT/UNL) pela oportunidade de participar neste projecto com bolsa de investigação.

Gostaria de agradecer à minha família, em particular aos meus pais, José e Luísa Ricardo, por todo o amor, dedicação, apoio incondicional e compreensão. Sem eles nada disto teria sido possível. O meu muito obrigado e eterna gratidão.

Gostaria também de agradecer a todos os meus amigos e colegas que de uma forma ou outra contribuíram para a minha formação académica, em particular ao Cristiano Lopes e ao Guilherme Silva.

Finalmente, o meu obrigado a todos aqueles que tiveram a disponibilidade e a amabilidade de me ajudar durante as avaliações das aplicações desenvolvidas.

A todos, muito obrigado!

Resumo

Actualmente existe uma necessidade crescente, por parte de museus e bibliotecas, em acompanhar as tendências tecnológicas dos nossos tempos. Estas instituições sentem necessidade em adaptar-se às novas tecnologias e assim inovar e desenvolver novas formas de interacção para atrair mais visitantes, sobretudo o público mais jovem, oferecendo experiências ricas, graças a um processo de aprendizagem e interacção mais lúdico.

Na sequência de um estudo sobre a cor em iluminuras medievais Portuguesas, surgiu a oportunidade de oferecer ao visitante uma experiência interactiva que contribua para a divulgação da arte ancestral das iluminuras, através da exploração de tecnologias de interacção computacional, de uma forma envolvente e lúdica, com a integração do real e do virtual.

O objectivo deste trabalho é assim desenvolver uma instalação interactiva que permita a criação de réplicas de iluminuras medievais, abordando vários aspectos, desde a origem das tintas e métodos de preparação, até ao processo de pintura propriamente dito, e que permita a partilha de informação, para o público em geral, sobre o contexto histórico e social da época, assim como o significado das cores e imagens usadas.

A instalação interactiva integrará três plataformas distintas. Uma aplicação desenvolvida para Tablet PCs, que permita simular todo o processo envolvido na criação de iluminuras, no período medieval, assim como a partilha de informação relevante sobre o processo. Uma aplicação desenvolvida para uma superfície interactiva multitoque de grandes dimensões, para exploração das iluminuras medievais e do seu contexto histórico e social. Finalmente, uma aplicação a ser desenvolvida com uma interface tangível, para a exploração de códices virtuais.

No âmbito desta dissertação encontra-se o desenvolvimento do *software* e mecanismos de interacção associados às duas primeiras aplicações, assim como a produção de um protótipo inicial para aferir a viabilidade da terceira aplicação.

Palavras-chave: Interfaces tangíveis, Interfaces multitoque, Interfaces caligráficas, Iluminuras medievais, Tecnologia e museus...

Abstract

Currently there is a growing need of museums and libraries to introduce new technological solutions. These institutions feel the need to adapt to new technologies, innovate and develop new forms of interaction to attract more visitors, especially younger audiences, offering rich experiences, through a fun and interactive learning process.

In the scope of a study about color in medieval Portuguese illuminations, the opportunity arose to offer the visitor an interactive experience that contributes to the dissemination of the ancient art of illuminations, by exploiting innovative interfaces, in an engaging and entertaining way, with the integration of real objects and virtual content.

The aim of this thesis is to develop an interactive installation that allows the creation of replicas of medieval illuminations, covering various aspects. It provides information, for the general public, about the historical and social context of the time, as well as the meaning of colors and images used.

The interactive installation will include three different platforms. An application developed for Tablet PCs, which allows simulating the whole process involved in creating illuminations in medieval times, as well as sharing relevant information about the process. An application developed for an interactive multi-touch surface of large dimensions, for exploration of medieval illuminated manuscripts and their historical and social context. Finally, an application to be developed with a tangible interface for virtual exploration of medieval codices.

Included in this thesis is the development of software and mechanisms of interaction associated with the first two applications, as well as the production of an initial prototype to assess the viability of the third application.

Keywords: Tangible interfaces, Multi-touch interfaces, Pen-based interfaces, Medieval illuminated manuscripts, Museums and technology. . .

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Descrição e contexto	2
1.3	Solução apresentada	3
1.4	Principais contribuições	5
1.5	Estrutura do relatório	6
2	Trabalho relacionado	7
2.1	Interfaces multitoque	7
2.1.1	Mesa multitoque para interacção multimédia em museus	12
2.1.2	Reactable	13
2.1.3	Memtable	15
2.2	Interfaces caligráficas	16
2.2.1	Teddy	18
2.2.2	CrossY	20
2.3	Sistemas para a exploração de manuscritos iluminados	21
2.3.1	Rerum Novarum	22
2.3.2	Turning the Pages™	22
2.3.3	Aplicação Web dedicada à produção de manuscritos iluminados	23
2.4	Considerações finais	24
3	Exploração de iluminuras medievais	27
3.1	Scriptorium Virtual	27
3.1.1	Produção de iluminuras	30
3.1.2	Interacção	31
3.1.3	Tecnologias utilizadas	40
3.2	Painel interactivo para exploração de iluminuras medievais	41
3.2.1	Consulta e exploração de iluminuras medievais	42
3.2.2	Interacção	43

3.2.3	Tecnologias utilizadas	49
4	Realização do sistema	51
4.1	Scriptorium Virtual	51
4.1.1	Qt	51
4.1.2	Arquitectura	55
4.1.3	Colecção de iluminuras	60
4.1.4	Seleccção de iluminuras	61
4.1.5	Produção de iluminuras	64
4.1.6	Preparação de tintas	71
4.2	Painel interactivo para exploração de iluminuras medievais	76
4.2.1	Arquitectura	76
4.2.2	Interacção multitoque	79
4.2.3	Carregamento de dados	81
4.2.4	Códices virtuais	82
5	Avaliação	89
5.1	Scriptorium Virtual	89
5.1.1	Observação de utilizadores	89
5.1.2	Inquérito aos utilizadores	91
5.2	Painel interactivo para exploração de iluminuras medievais	97
6	Conclusões e trabalho futuro	99
6.1	Conclusões	99
6.2	Trabalho futuro	100
A	Questionários	107
B	Resultados dos inquéritos	115
B.1	Scriptorium Virtual	116
B.2	Painel interactivo para exploração de iluminuras medievais	118
C	Formulário de observação	121
D	Resultados da observação	127
E	XML Schemas	131

Lista de Figuras

1.1	Protótipo do códice virtual	4
1.2	Modelo tridimensional controlado pelo protótipo	5
2.1	Transição para um novo paradigma de interacção	8
2.2	Exemplo de mesa multitoque	10
2.3	Mesa interactiva multitoque	12
2.4	Interface da mesa interactiva	13
2.5	Reactable	14
2.6	Arquitectura Reactable	15
2.7	A Memtable	16
2.8	Teddy	19
2.9	CrossY	20
2.10	Rerum Novarum	22
2.11	Turning the Pages™	23
2.12	Escrita e decoração de manuscritos	24
3.1	Janela de selecção de iluminuras	33
3.2	Janela de produção de iluminuras	34
3.3	Iluminura original vs. iluminura digital	34
3.4	Utilizador a pintar iluminura	35
3.5	Janela de preparação de tintas	38
3.6	Janelas com informações sobre as iluminuras	40
3.7	HP Compaq tc4200 Tablet PC	41
3.8	Menu inicial	45
3.9	Menu «Portugal Românico»	45
3.10	Exemplos de janelas de «Portugal Românico»	46
3.11	Menu de selecção de cor e menu da cor escolhida	46
3.12	Exemplos de janelas das cores	47
3.13	Exemplo de códice virtual	48

3.14	Janela «Scriptorium»	48
3.15	Janela «Códice medieval»	49
3.16	Galeria de iluminuras	50
4.1	Visão geral da biblioteca Qt	54
4.2	Arquitectura da aplicação Scriptorium Virtual	56
4.3	<i>Layout</i> da classe QMainWindow	57
4.4	Diagrama de classes da aplicação Scriptorium Virtual	61
4.5	Máquina de estados associada à preparação da cor azul	73
4.6	Arquitectura do painel interactivo para exploração de iluminuras medievais	78
4.7	Protocolo TUIO	79
4.8	Diagrama de classes da aplicação multitoque	82
4.9	Linha de simetria	83
4.10	As mudanças que ocorrem quando uma página é virada	84
4.11	Movimento da linha de simetria	85
4.12	Moldura da área c	87
5.1	Resultados relativos à facilidade de aprendizagem e usabilidade da aplicação	93
5.2	Resultados relativos à interacção dos utilizadores com a aplicação	94
5.3	Resultados relativos à facilidade de execução de determinadas operações	95
5.4	Resultados relativos ao envolvimento emocional dos utilizadores com a aplicação	95
E.1	XML Schema que define as regras de validação da aplicação Scriptorium Virtual	132
E.2	XML Schema que define as regras de validação das janelas do painel interactivo	133

Listagens

4.1	Função main da aplicação Scriptorium Virtual	55
4.2	Criar animações em Qt	57
4.3	Inicialização e configuração da aplicação Scriptorium Virtual	59
4.4	<i>Thumbnail</i> de iluminura especificado no ficheiro XML	62
4.5	Passo da produção de iluminuras especificado no ficheiro XML	65
4.6	Método para calcular a desaceleração associada ao efeito <i>kinetic scrolling</i>	67
4.7	Passo da preparação de tintas especificado no ficheiro XML	72
4.8	Interface iPanel	77
4.9	Inicialização e configuração da aplicação multitoque	78
4.10	Métodos de tratamento de eventos TUIO	80
4.11	Inicialização e configuração de cliente TUIO	80



Introdução

Os constantes avanços tecnológicos, a crescente facilidade no acesso à tecnologia e o emergente entusiasmo em relação a novos paradigmas de interacção, sobretudo por parte do público mais jovem, incentivam cada vez mais o desenvolvimento de novas soluções interactivas, que permitam interagir e visualizar informação de forma mais eficaz e satisfatória.

A tendência passa pelo desenvolvimento de plataformas interactivas ajustadas às diferentes realidades aplicacionais, que ofereçam um acesso directo à informação disponível através de experiências de interacção mais naturais e atractivas para os seus utilizadores. Este tipo de soluções é procurado cada vez mais por instituições em busca de meios inovadores para transmitirem as suas mensagens e, sobretudo, responderem às expectativas de um público cada vez mais exigente.

O sucesso destas soluções depende principalmente da eficácia da transmissão da mensagem que, por sua vez, se encontra intrinsecamente relacionada com a forma como esta é transmitida. O desafio passa por encontrar um equilíbrio entre conteúdo e forma, de modo a desenvolver uma plataforma interactiva que cumpra os requisitos, mas simultaneamente proporcione uma experiência agradável, atractiva e eficaz aos seus utilizadores.

1.1 Motivação

Instituições, como museus e bibliotecas, tentam cada vez mais acompanhar o rápido progresso tecnológico, desenvolvendo novas plataformas de interacção e exploração dos seus conteúdos, de forma a atrair mais visitantes, sobretudo o público mais jovem, e ir ao encontro do que estes procuram, oferecendo experiências mais ricas e atractivas, ao mesmo tempo que confrontam os visitantes com os conteúdos de uma forma mais lúdica.

Na sequência de um estudo sobre a cor em iluminuras medievais Portuguesas, levado a

cabo pelo Departamento de Conservação e Restauro da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, surgiu a intenção de divulgar os resultados obtidos e a arte ancestral das iluminuras medievais ao público em geral, em especial a crianças e ao público mais jovem, recorrendo a uma plataforma interactiva e à exploração das mais recentes tecnologias de interacção computacional. O objectivo passa por desenvolver um sistema interactivo fácil de usar e que envolva o utilizador numa experiência rica, tanto do ponto de vista informativo, como do ponto de vista lúdico. As tarefas e objectivos propostos por este projecto incentivam à inovação e contribuem para um melhor acesso e para a conservação das iluminuras medievais.

Para além das motivações que dizem respeito à forma e conteúdo, existem também motivações científicas e tecnológicas para o desenvolvimento desta instalação interactiva. Os desafios apresentados por este tipo de sistemas apresentam a oportunidade perfeita para a exploração das mais recentes técnicas de interacção e acesso à informação. Surge assim a oportunidade, dentro do contexto deste projecto, de explorar e criar novas formas de interacção, que permitam navegar facilmente nas diferentes interfaces desenvolvidas e, simultaneamente, ofereçam uma experiência mais natural, enriquecedora e recreativa aos seus utilizadores.

1.2 Descrição e contexto

O conceito desta instalação interactiva nasce na sequência de um conjunto de *workshops* realizados pelo Departamento de Conservação e Restauro, onde, para além da divulgação do tema das iluminuras medievais conjuntamente com o seu contexto histórico e social, são preparadas tintas e produzidas iluminuras, recorrendo a materiais e técnicas semelhantes às originais, usadas no *scriptorium*¹ medieval. A solução proposta pretende simular todos os processos envolvidos nestes *workshops*, oferecendo aos utilizadores a oportunidade de criarem réplicas de iluminuras medievais, ao mesmo tempo que são confrontados com todos os pormenores relacionados com a sua execução, desde a origem das tintas e métodos de preparação, até ao processo de pintura propriamente dito. O utilizador poderá depois partilhar as suas iluminuras com os demais utilizadores, proporcionando uma experiência de aprendizagem rica, que incitará ao aperfeiçoamento das iluminuras criadas e, conseqüentemente, fomentará o interesse pelas iluminuras e por todo o processo criativo.

Para além do processo de aprendizagem envolvido na criação de iluminuras, um dos principais objectivos desta solução passa por disponibilizar ao público em geral um meio de acesso privilegiado ao tema das iluminuras medievais, um local onde o utilizador possa facilmente consultar todas as informações disponíveis sobre o tema de um ponto de vista científico, social, artístico e histórico. A instalação interactiva incluirá ainda a exploração virtual de códices medievais completos, experiência que permitirá ao utilizador, não só consultar obras que em condições normais não estariam ao seu alcance, mas também adquirir conhecimentos gerais sobre os códices consultados.

O recurso a plataformas interactivas é fundamental no contexto deste projecto pois evita

¹Uma divisão nos mosteiros medievais europeus destinada aos monges copistas que, na época medieval, escreviam os manuscritos.

o contacto directo com as obras originais, contribuindo para a sua conservação e, simultaneamente, permitindo o seu estudo e divulgação. Deseja-se que esta instalação interactiva contribua para essa divulgação e que permita aos visitantes ficar com uma ideia concreta sobre o que são as iluminuras medievais, o que representam, os diferentes aspectos associados à sua concepção e o seu contexto histórico e social.

Pretende-se que estas funcionalidades sejam disponibilizadas numa plataforma desenvolvida a pensar no utilizador, que permita uma interacção natural e envolvente, e que facilite o processo de aprendizagem e divulgação das iluminuras medievais. Esta instalação interactiva poderá ser facilmente implementada, por exemplo, em locais de interesse cultural, enriquecendo uma exposição ou até nas instituições onde as obras são preservadas.

O trabalho desta dissertação insere-se no âmbito do projecto «Colour in medieval illuminated manuscripts: between beauty and meaning», levado a cabo pelo Departamento de Conservação e Restauro (DCR) em parceria com o Centro de Informática e Tecnologias de Informação (CITI), ambos da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT/UNL), com o apoio financeiro da Fundação para a Ciência e a Tecnologia do Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES). Este projecto surgiu na sequência de um estudo interdisciplinar sobre a cor em iluminuras medievais Portuguesas, iniciado em 2005 pelo DCR, que conta com a colaboração de um grupo multidisciplinar composto por historiadores, bibliotecários, químicos, conservadores e curadores. Esta dissertação consiste essencialmente na produção da instalação interactiva, acima descrita, para divulgação dos resultados obtidos neste estudo e do tema das iluminuras em geral.

1.3 Solução apresentada

Como referido anteriormente, o objectivo desta dissertação consiste no desenvolvimento de uma instalação interactiva que contribua para a divulgação das iluminuras medievais através da exploração das mais recentes tecnologias de interacção computacional e que ofereça uma experiência envolvente e lúdica aos seus utilizadores. A solução apresentada é constituída por três aplicações distintas, cada uma com o seu propósito e pensadas de raiz para técnicas de interacção que maximizem a experiência do utilizador. A presente dissertação contempla o desenvolvimento de duas destas aplicações, assim como o desenvolvimento de um protótipo para avaliar a viabilidade da terceira aplicação dentro do contexto do projecto.

A instalação interactiva inclui uma aplicação que ofereça ao utilizador uma experiência individual de produção de iluminuras, similar à oferecida pelo *workshop*, mas desta feita numa plataforma digital. Esta aplicação será desenvolvida e optimizada para Tablet PCs, assente no paradigma de interfaces caligráficas com caneta, tendo como principais objectivos recriar o processo de produção de iluminuras e a partilha de informação que seja relevante durante o processo. A aplicação permitirá que o utilizador experiencie o processo de produção de iluminuras usando um Tablet PC e uma caneta, ao invés de pergaminho e pincéis como no *workshop*, acompanhando-o passo-a-passo com informações pertinentes sobre o que fazer a cada momento. A recriação de alguns aspectos associados ao *workshop* e subjacentes à actividade de

pintar e desenhar, como variações na espessura e tipo de traço ou mesmo o consumo da tinta usada na caneta, permitirá envolver mais o utilizador na experiência de produção de iluminuras e abstraí-lo da plataforma digital usada. Terminado o processo de criação, a iluminura é armazenada, ficando disponível para partilha com os demais utilizadores. Tal como no *workshop*, o utilizador poderá experienciar também o processo de preparação das tintas utilizadas na criação das iluminuras. O utilizador poderá moer e misturar os materiais necessários à preparação da tinta escolhida, simulando os procedimentos reais, à medida que é guiado passo-a-passo com informações relevantes sobre o que fazer a cada momento. A aplicação compreende ainda uma componente informativa, onde o utilizador poderá consultar informação relacionada com a iluminura que está a reproduzir no momento, como por exemplo, o simbolismo dos diferentes elementos decorativos e cores presentes na iluminura e o significado dos textos apresentados.

A instalação interactiva inclui outra aplicação a ser desenvolvida para uma superfície interactiva multitoque de grandes dimensões, construída em torno da temática da cor em iluminuras medievais Portuguesas, mas que ofereça também uma visão geral sobre o tema, quer do ponto de vista científico e social, quer do ponto de vista artístico e histórico. Pretende-se que esta aplicação reúna toda a informação existente sobre o tema num só local, oferecendo aos visitantes acesso à informação de uma forma simples, organizada e que encoraje a exploração e aprendizagem. Esta aplicação integra ainda uma galeria com as iluminuras criadas pelos utilizadores na aplicação apresentada anteriormente e a possibilidade de consultar e explorar versões digitais de códices medievais.

Na fase inicial do trabalho foi ainda desenvolvido um protótipo para avaliar a viabilidade de uma terceira aplicação dentro do contexto do projecto. Pretende-se que esta terceira aplicação permita explorar códices medievais através da manipulação de objectos físicos. A aplicação será desenvolvida com o suporte de um objecto físico equipado com sensores que detectam a sua posição, num paradigma nomeadamente designado por interfaces tangíveis, com a integração do real e do virtual. Através deste objecto, neste caso uma reprodução de um códice medieval Português, o utilizador será capaz de controlar um modelo tridimensional, representativo do códice em questão. Ao mover este modelo o utilizador poderá examinar em detalhe a capa, contracapa e lombada do códice, surgindo também informações visuais e textuais relacionadas com os materiais e técnicas usadas na sua concepção e sobre alguns aspectos relevantes sobre o seu conteúdo. A aplicação prevê ainda que o utilizador abra e folheie o códice, de modo a consultar os fólios que o constituem.

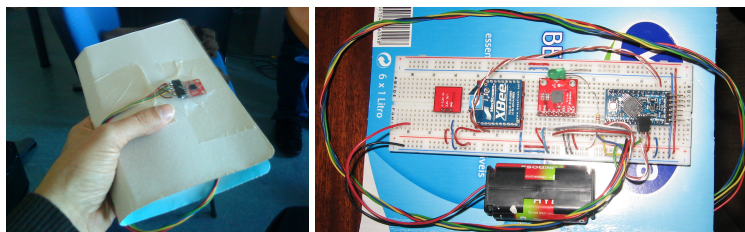


Figura 1.1: Protótipo do códice virtual

O objectivo do protótipo desenvolvido no âmbito desta dissertação consiste em averiguar que sensores serão necessários para detectar correctamente os movimentos de um objecto físico (neste caso um livro) no espaço tridimensional. O protótipo desenvolvido (figura 1.1) é composto por dois acelerómetros de três eixos (um para a capa e outro para a contracapa), um controlador Arduino e duas antenas Wi-Fi (uma emissora, acoplada ao livro, e outra receptora ligada via USB ao computador a correr a aplicação). O *hardware* deste protótipo foi desenvolvido fora do âmbito desta dissertação, ficando o autor deste documento responsável apenas pelo desenvolvimento do *software*. Este *software* consiste numa aplicação que recebe as acelerações produzidas pelos acelerómetros numa porta série virtual, convertendo-as em ângulos, que depois são aplicados a um modelo tridimensional simples de um livro (figura 1.2). Apesar das diferentes experiências realizadas e de se ter conseguido uma aproximação razoável, os movimentos do protótipo apresentaram-se sempre com limitações impossíveis de ultrapassar. Chegou-se, portanto, à conclusão que os sensores usados são insuficientes para o pretendido, encontrando-se em desenvolvimento um novo sistema, fora do âmbito desta dissertação, com mais sensores, nomeadamente, um giroscópio e um magnetómetro, que conjuntamente com os acelerómetros permitem de facto detectar correctamente os movimentos de um objecto no espaço tridimensional.

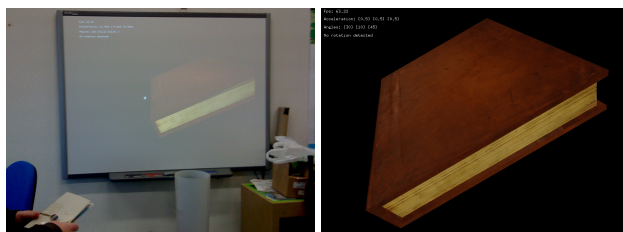


Figura 1.2: Modelo tridimensional controlado pelo protótipo

As duas aplicações desenvolvidas no âmbito desta dissertação podem trabalhar de forma independente ou como uma instalação completa, podendo inclusive no futuro serem acompanhadas pela terceira aplicação, completando assim a instalação interactiva proposta e oferecendo a melhor experiência possível aos seus utilizadores.

1.4 Principais contribuições

As principais contribuições, para a solução apresentada, são:

- Desenvolvimento de uma aplicação para Tablet PCs que permita simular o processo envolvido na criação de iluminuras, assim como a divulgação de informação sobre as mesmas.
- Desenvolvimento de uma aplicação para superfície interactiva multitoque de grandes dimensões, que reúna num só local toda a informação disponível sobre o tema e simultaneamente ofereça uma experiência informativa e lúdica aos seus utilizadores.

- Desenvolvimento de um protótipo para avaliar a viabilidade de uma aplicação baseada no conceito de interfaces tangíveis, dentro do contexto do projecto, para a exploração de códigos medievais.
- Colaboração na produção e integração de conteúdos nas diferentes aplicações desenvolvidas.
- Avaliação das aplicações desenvolvida com utilizadores e peritos.

O desenvolvimento destas aplicações engloba a criação de interfaces de dois paradigmas de interacção diferentes, assim como a lógica associada ao seu funcionamento.

1.5 Estrutura do relatório

Este documento encontra-se estruturado em seis capítulos, apresentados de seguida.

Capítulo 1 - Introdução: Este capítulo apresenta uma visão global da dissertação, o seu contexto, as suas motivações, uma descrição clara do problema e ainda a solução apresentada, assim como as principais contribuições para o seu desenvolvimento.

Capítulo 2 - Trabalho relacionado: Este capítulo apresenta o estado da arte em relação às tecnologias incluídas no âmbito desta dissertação, nomeadamente, interfaces caligráficas em Tablet PCs e interfaces multitoque. São ainda apresentados alguns exemplos da adaptação destas técnicas ao tema dos manuscritos iluminados.

Capítulo 3 - Exploração de iluminuras medievais: Neste capítulo apresentam-se as aplicações desenvolvidas no âmbito desta dissertação. Para cada uma das aplicações são apresentados em detalhe os conceitos considerados relevantes, as tecnologias utilizadas no seu desenvolvimento, assim como as suas funcionalidades e modos de interacção.

Capítulo 4 - Realização do sistema: Neste capítulo apresentam-se os aspectos considerados mais relevantes da produção da instalação interactiva desenvolvida no âmbito desta dissertação. São ainda apresentadas as dificuldades encontradas durante a realização do sistema e as soluções implementadas como resposta.

Capítulo 5 - Avaliação: Neste capítulo apresentam-se e discutem-se os resultados obtidos nas avaliações feitas às aplicações desenvolvidas no âmbito desta dissertação.

Capítulo 6 - Conclusões e trabalho futuro: Neste capítulo é feita uma análise do trabalho realizado no âmbito desta dissertação e é apresentado o trabalho a desenvolver futuramente para a evolução da instalação interactiva.



Trabalho relacionado

Neste capítulo apresenta-se um resumo do estado da arte em relação às técnicas de interacção abordadas nesta dissertação. Através da enumeração de vários exemplos relacionados, apresentam-se alguns conceitos e técnicas necessárias para o desenvolvimento da solução proposta.

2.1 Interfaces multitoque

Segundo Ishii e Ullmer [IU97], graças à crescente evolução das tecnologias digitais e à crescente utilização dos computadores pessoais em residências, empresas e escolas, os seres humanos estão a desperdiçar grande parte da sua capacidade preênsil para agarrar e manipular objectos. Segundo os autores, vivemos numa era em que os seres humanos estão ligados em simultâneo a duas realidades paralelas: o ambiente físico e o ambiente digital. As razões apontadas para a separação destas duas realidades são as limitações subjacentes à tradicional *graphical user interface* (GUI). Apesar do ser humano ter desenvolvido um sem número de capacidades e práticas de trabalho para processar informação recorrendo à manipulação de objectos físicos, estas são desperdiçadas pelos actuais paradigmas de interacção pessoa-máquina devido à falta de diversidade dos meios *input/output*.

As interfaces gráficas tradicionais, que estamos habituados a utilizar diariamente, são essencialmente definidas por uma série de elementos gráficos (e.g., janelas, ícones, menus) que existem num estado puramente electrónico, ou seja, existem apenas virtualmente. Estes elementos podem ser manipulados através de dispositivos genéricos (e.g., rato e teclado) que, devido à sua natureza, apresentam uma série de limitações. Uma das limitações, identificada por Fitzmaurice, Ishii e Buxton [FIB95], subjacente ao paradigma das interfaces gráficas e seus dispositivos *input/output*, é o facto de apenas permitirem uma multiplexagem temporal para

input, ou seja, através de um único dispositivo (e.g., rato) é possível controlar diferentes funções em diferentes instantes, e uma multiplexagem espacial para *output*, ou seja, os ícones, janelas e menus ocupam o seu próprio espaço e precisam de estar visíveis para serem usados. Esta dissonância entre uma multiplexagem temporal para *input* e uma multiplexagem espacial para *output* faz com que apenas uma tarefa seja executada a cada instante. Nasceu assim a motivação de evoluir para além do modelo dominante das GUI e criar um novo paradigma de interacção, onde o mundo físico seria aumentado através da incorporação de informação digital nos objectos físicos do nosso dia-a-dia, resultando numa experiência mais natural e numa interacção uniforme entre utilizador e informação. Aproveitando a capacidade natural dos seres humanos para manipular objectos e a sua sensibilidade ao meio que os rodeia, passa a ser possível criar um modelo interactivo mais produtivo e natural para o utilizador. A figura 2.1 ilustra a transição de um paradigma de interacção pessoa-máquina assente nas GUI, usadas abundantemente em Desktop PCs, para o novo paradigma das interfaces tangíveis, que transformarão o mundo numa interface.

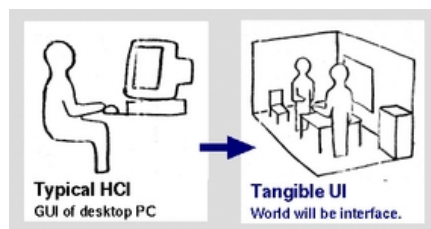


Figura 2.1: Transição para um novo paradigma de interacção

As *tangible user interfaces* (TUIs) representam uma nova forma de incorporar a visão de computação ubíqua introduzida por Mark Weiser [Wei91] em 1991, através da integração de tecnologias digitais no ambiente físico, tornando a tecnologia invisível para o utilizador.

Ishii e Ullmer [IU97] identificam e apresentam protótipos para três conceitos distintos de interfaces tangíveis:

Ambient Media: Consiste no recurso à percepção periférica do ser humano para a transmissão de informação secundária, mas que possa ser relevante, através do som, luz, ar, água, etc. O protótipo apresentado para este conceito, denominado *ambientROOM*, serve como plataforma complementar e permite, por exemplo, associar um *phicon* (*physical icons* ou ícones físicos, são objectos físicos que representam uma determinada informação digital) a uma determinada página Web, traduzindo-se cada acesso a essa página no som de gotas de água. Indica assim o som de chuva forte, uma elevada frequência de acessos, e a ausência de som precisamente o contrário. Desta forma o utilizador poderá estar concentrado noutro tipo de tarefas e em simultâneo, recorrendo à sua percepção periférica, manter-se informado sobre determinado assunto.

Coupling of Bits and Atoms: Incorporação de informação digital em objectos do nosso dia-a-dia, susceptíveis de serem manipulados (e.g., cartões, livros, ferramentas). Este conceito é considerado o mais tradicional na área das interfaces tangíveis e é derivado do conceito

de *graspable user interfaces* (interfaces palpáveis do utilizador) introduzido por Fitzmaurice, Ishii e Buxton [FIB95] em 1995, tendo sido inclusive esse o nome usado inicialmente para as interfaces tangíveis. O protótipo apresentado por Ishii e Ullmer, denominado metaDESK, é constituído essencialmente por uma superfície gráfica retroiluminada, um monitor LCD montado num braço mecânico, denominado activeLENS, e uma série de *phicons* (representados por sólidos geométricos). Este protótipo permite aumentar a realidade dos objectos colocados sobre a mesa, quando observados através da activeLENS. Um exemplo aplicacional apresentado para esta mesa é o Tangible Geospace. Esta aplicação permite que os utilizadores manipulem mapas 2D e 3D do campus do MIT, exibidos na superfície, através do posicionamento e manipulação de *phicons* sobre a mesa. Estes *phicons*, representativos de edifícios do MIT, surgem, não como simples sólidos geométricos, mas como modelos tridimensionais dos edifícios em questão. Com o uso de dois *phicons* em simultâneo, representando dois edifícios, é também possível aplicar transformações de escala no modelo, para além de translações e rotações.

Interactive surfaces (superfícies interactivas): A transformação de todas as superfícies disponíveis num dado espaço arquitectónico (e.g., paredes, mesas, tectos, portas, janelas) em interfaces activas entre o mundo físico e o virtual. O protótipo apresentado para este conceito, denominado transBOARD, consiste num quadro branco normal, onde os utilizadores podem escrever e desenhar, mas que permite absorver a informação que é escrita fisicamente e transformá-la em informação digital, ficando esta disponível na rede.

É neste último conceito de superfícies interactivas que assenta uma das aplicações a ser desenvolvida no âmbito desta dissertação, uma aplicação que possa ser implementada numa superfície interactiva com uma interface multitoque acessível a todos os utilizadores.

As interfaces multitoque permitem ao utilizador interagir com informação digital através de superfícies sensíveis ao toque. Apesar de usada recorrentemente para descrever outras implementações mais limitadas, como o *dual-touch*, a nomenclatura multitoque refere-se à capacidade destes dispositivos de registar simultaneamente três ou mais posições distintas sobre uma superfície. Esta capacidade, associada à elevada dimensão da maioria das superfícies, permite uma interacção colaborativa, com vários utilizadores em simultâneo e com detecção de objectos tangíveis sobre a superfície (interface tangível).

Originalmente associadas a superfícies de elevada dimensão, as interfaces multitoque começam a tornar-se do domínio comum, graças à crescente proliferação de dispositivos de dimensão reduzida que integram este tipo de tecnologia (e.g., Smartphones, PDAs, Tablet PCs) e sobretudo aos avanços tecnológicos que se têm verificado nesta área. Estes avanços permitem evitar as dispendiosas soluções comerciais e construir, com relativa facilidade, soluções de qualidade recorrendo a técnicas e materiais acessíveis. Nesta área de investigação é importante destacar o contributo fundamental de Jeff Han [Han05], com a apresentação da sua solução para sensores multitoque de baixo custo, conhecida como Frustrated Total Internal Reflection (FTIR) por tirar partido de um fenómeno com o mesmo nome.

Esta solução, assim como todas as relacionadas, baseiam-se no mesmo princípio, na análise

e detecção de irregularidades capturadas por uma câmara sensível à luz infravermelha. Isto é conseguido da seguinte forma: uma ou mais fontes de luz infravermelha projectam um espectro contínuo e uniforme de luz na superfície interactiva; sempre que algo entra em contacto com esta superfície provoca uma anomalia no espectro; este distúrbio na reflexão da luz infravermelha é capturado graças a um sensor, normalmente uma câmara sensível à luz infravermelha, e a um computador que recorre a *software* de processamento de imagem para detectar a posição e a dimensão do distúrbio; esta informação é depois processada dentro do seu contexto e transmitida ao utilizador através de um projector montado para o efeito (figura 2.2).

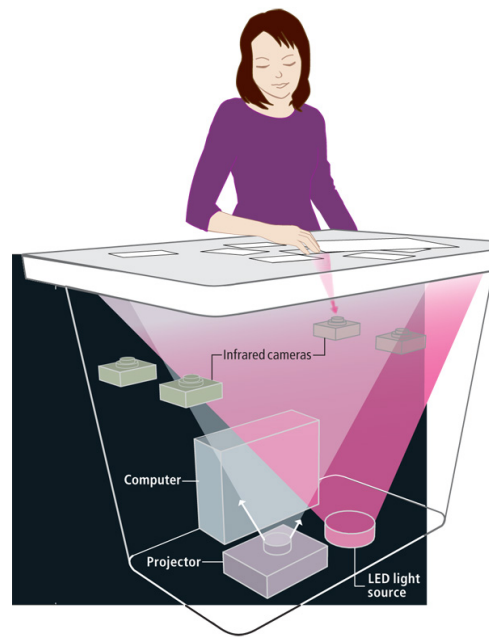


Figura 2.2: Exemplo de mesa multitoque

Depois da solução apresentada por Jeff Han, começaram a surgir uma série de variantes desta tecnologia, cada uma com as suas vantagens e desvantagens. Entre as mais divulgadas encontra-se a Diffused Illumination (DI), usada pela Microsoft na primeira versão da Microsoft Surface e a Laser Light Plane (LLP), desenvolvida pela comunidade NUI Group [Gro11] como sendo uma das soluções mais fáceis e baratas de implementar.

Apesar da existência de uma série de variantes desta tecnologia, é um facto que as diferenças subsistem sobretudo no posicionamento das fontes de luz infravermelha e nos fenómenos resultantes desse posicionamento sempre que há contacto com a superfície. Em tudo o resto são bastante semelhantes: sistemas constituídos por uma superfície, iluminada por uma ou mais fontes de luz infravermelha, um computador, onde são utilizadas técnicas de processamento de imagem, capturadas por uma câmara sensível à luz infravermelha, e um projector.

Estas semelhanças e, sobretudo, as facilidades introduzidas pela solução apresentada por Jeff Han, levaram a que comunidades independentes desenvolvessem *software* de suporte a interfaces multitoque, são exemplos disso a infraestrutura reacTIVision [KB07], a plataforma TUIO [KBBC05], a biblioteca Touchlib [D.11] e a solução Community Core Vision (CCV) [z111].

O reacTIVision é um sistema *open source*, multiplataforma, que oferece mecanismos robustos para detectar rapidamente o toque de múltiplos dedos em simultâneo e detectar e reconhecer objectos físicos com marcadores *fiducial*. Esta detecção é feita através da análise contínua de uma sequência de vídeo produzida por uma câmara sensível à luz infravermelha. Sempre que resulta desta análise a detecção de um distúrbio, seja devido a um objecto em particular ou um dedo, é comunicado esse evento à camada aplicacional. A comunicação é feita recorrendo ao protocolo TUIO, que facilita a codificação e transmissão relativa aos eventos detectados.

A plataforma TUIO, implementada pela primeira vez durante o desenvolvimento do reacTIVision, define um protocolo comum e uma API (*Application Programming Interface*) para superfícies multitoque. Este protocolo permite a transmissão de representações abstractas de superfícies interactivas, incluindo os eventos de toque e o estado de objectos tangíveis, entre a aplicação responsável por interpretar as imagens produzidas pela câmara sensível à luz infravermelha e as camadas aplicacionais responsáveis por processar o estado da superfície e dar *feedback* visual aos utilizadores. As representações são transmitidas como mensagens UDP (*User Datagram Protocol*), garantindo uma comunicação de baixa latência. O protocolo TUIO é codificado usando o formato *Open Sound Control* (OSC) [WFM03], que fornece um método de codificação binário para a transmissão de mensagens entre computadores, sintetizadores de som e outros dispositivos multimédia. Assim, o protocolo TUIO pode ser facilmente implementado em qualquer plataforma que suporte o formato OSC.

Criada inicialmente para a técnica Frustrated Total Internal Reflection (FTIR), a Touchlib é uma biblioteca *open source* para a criação de superfícies interactivas multitoque. A biblioteca Touchlib oferece, entre outras funcionalidades, a detecção em tempo real de *blobs* e eventos multitoque de uma sequência de vídeo produzida por uma câmara sensível à luz infravermelha, recorrendo à biblioteca OpenCV (*Open Source Computer Vision*) [Ope11]. A Touchlib funciona numa arquitectura cliente-servidor, com esta a assumir o papel de servidor e a enviar aos clientes na camada aplicacional, mensagens TUIO com a codificação dos eventos detectados.

A solução Community Core Vision (CCV), também conhecida como tbeta, é uma aplicação *open source*, multiplataforma, de visão por computador, que permite a construção de sistemas multitoque baseados em várias técnicas de iluminação. A CCV suporta, entre outras, todas as técnicas apresentadas anteriormente, nomeadamente, FTIR, DI e LLP. A solução CCV recebe como *input* dados de vídeo capturados em tempo real por uma câmara digital ou previamente armazenados num ficheiro de vídeo, analisa os dados em busca de *blobs* e eventos multitoque, e depois comunica-os à camada aplicacional através de TUIO/OSC/XML.

Para além da tecnologia de infravermelhos, ideal para soluções de baixo custo e grandes superfícies, existem outras implementações, normalmente encontradas em dispositivos móveis de pequena dimensão, para a construção de superfícies multitoque:

Resistivo: É uma técnica que utiliza duas folhas flexíveis revestidas com um material resistivo, posicionadas paralelamente e separadas por um reduzido espaço. Sempre que é feito contacto com a superfície as duas folhas tocam-se. Nessas duas folhas existem linhas horizontais e verticais que quando estão juntas permitem registar a localização precisa

do toque.

Capacitivo: É uma tecnologia que recorre a uma superfície feita de um material isolante, como o vidro, revestida com um condutor transparente. Como o corpo humano é também um condutor, um toque na sua superfície provoca uma distorção no campo electrostático, mensurável como uma variação na capacitância. Através desta variação é possível detectar a posição do toque com bastante precisão.

A Microsoft lançou recentemente uma nova versão da sua mesa interactiva, a Microsoft Surface 2.0 [z211]. Esta nova versão, entre outras novidades, apresenta uma tecnologia inovadora, designada por PixelSense™, onde cada *pixel* da superfície interactiva actua também como uma câmara infravermelha. Esta nova tecnologia permite o uso de *displays* de alta definição, oferecendo uma qualidade de imagem sem precedentes em mesas desta dimensão. Graças às câmaras infravermelhas espalhadas por toda a superfície é possível ver exactamente o que se passa sobre esta, permitindo reconhecer todo o tipo de gestos, movimentos e objectos. Esta tecnologia permite, por exemplo, colocar uma folha de papel sobre a superfície interactiva e reconhecer rapidamente o texto desse documento. Ao contrário da versão anterior, que usava tecnologia de infravermelhos, esta nova versão não necessita de projectores, nem câmaras por baixo da superfície, permitindo uma construção muito mais fina e leve.

A seguir apresentam-se alguns exemplos de interfaces multitoque construídas a partir das técnicas apresentadas anteriormente, que serviram de inspiração no desenvolvimento da aplicação proposta.

2.1.1 Mesa multitoque para interacção multimédia em museus

Este projecto, desenvolvido no Centro de Informática e Tecnologias da Informação (CITI) da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, visava a criação de uma mesa multitoque para a exploração de uma colecção de arte [CMN⁺10]. A mesa foi instalada numa exposição de arte contemporânea, junto das obras representadas, com o objectivo de oferecer aos visitantes informações sobre as obras expostas, através de uma experiência interactiva rica e colaborativa, onde vários visitantes podem aceder ao sistema em simultâneo. Esta instalação interactiva explora a noção de um ambiente colaborativo, onde as obras de arte do museu podem ser revisitadas e exploradas, e a opinião e *feedback* do visitante pode ser gravado.



Figura 2.3: Mesa interactiva multitoque

A mesa multitoque (figura 2.3) é constituída por um projector, um sistema de espelhos, uma câmara PlayStation Eye (modificada para visão infravermelha), um fita montada em torno da superfície com LEDs infravermelhos, um computador portátil, colunas de som e uma superfície de grandes dimensões, acessível a vários utilizadores ao mesmo tempo. Foi usada esta configuração de forma a aplicar a solução Frustrated Total Internal Reflection, mencionada anteriormente, ideal para a construção de superfícies multitoque de baixo custo.

A aplicação foi desenvolvida em openFrameworks [LW11], uma biblioteca *open source* em C++, adequada ao desenvolvimento de aplicações criativas, com uma forte componente visual. O sistema utiliza a solução Community Core Vision (CCV) para a detecção de múltiplos eventos relativos à posição dos dedos na superfície da mesa e o protocolo TUIO para a combinação, codificação e transmissão de eventos de toque à camada aplicacional.

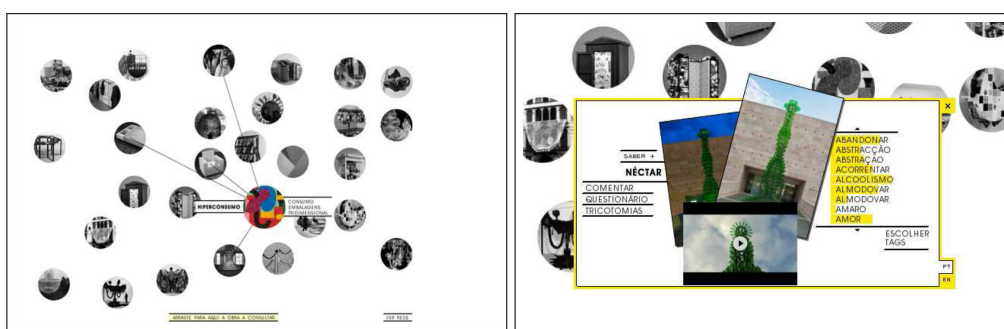


Figura 2.4: Interface da mesa interactiva

A interface (figura 2.4) da aplicação é constituída por uma série de imagens circulares a preto e branco, que representam as obras da colecção de arte, espalhadas por toda a interface, com um posicionamento inicial predefinido. As principais funcionalidades da aplicação são: navegar e explorar a colecção de arte, consultar informação das obras, preencher questionários, atribuir e navegar por *tags*, livro de visitas e jogos casuais. A interface inclui um botão que mostra toda a rede de ligações entre as obras de arte, resultante das relações entre as *tags*. Na parte inferior da interface existe uma área para onde o utilizador pode arrastar uma obra, surgindo uma janela com informações, incluindo imagens e vídeos, sobre a obra seleccionada. Podem também ser consultados ou adicionados comentários à obra em questão e podem ser atribuídas novas *tags* por parte dos visitantes. A interface também permite o preenchimento de um questionário sobre a exposição e sobre o uso de tecnologia no espaço de um museu, e inclui ainda um jogo sobre as relações semânticas entre as obras em exposição.

2.1.2 Reactable

O Reactable [JGAK07] é um instrumento musical electrónico, desenvolvido pela Reactable Systems em colaboração com o Music Technology Group, ambos da Universitat Pompeu Fabra, em Barcelona, que oferece uma interface tangível multitoque para múltiplos utilizadores em simultâneo, na superfície de uma mesa interactiva. A principal motivação para o desenvolvimento desta plataforma interactiva foi criar um instrumento musical atractivo para múltiplos

utilizadores, que permitisse a criação de música electrónica envolvendo o utilizador desde o primeiro minuto, mas que simultaneamente fosse complexo, subtil e oferecesse infinitas variações.



Figura 2.5: Reactable

A superfície interactiva do Reactable é uma mesa redonda translúcida em tons de azul (figura 2.5), usada sempre em ambientes com pouca iluminação, com o aspecto de um *display* retroiluminado que se ilumina com animações dinâmicas representativas das mudanças musicais. A colocação de blocos físicos sobre a mesa e a interacção com a apresentação visual via manipulação desses blocos ou pontas dos dedos, acciona um sintetizador virtual, resultando na criação de música ou efeitos sonoros. Nesta interacção cada bloco representa uma função diferente (e.g., criar som, filtrar som, controlo de parâmetros de áudio), desta forma a manipulação de cada bloco permite o controlo das diferentes funções. O facto de suportar multitoque numa superfície de elevadas dimensões incentiva a uma interacção colaborativa, com múltiplos utilizadores a participar em simultâneo neste processo criativo. O sistema permite ainda estender este conceito suportando o aumento da superfície interactiva através da ligação remota a outras mesas, partilhando todas o mesmo espaço virtual.

O Reactable apresenta uma configuração baseada numa variante da técnica *Diffused Illumination*, denominada *Rear Diffused Illumination*. O sistema é composto por uma câmara sensível à luz infravermelha, apontada à superfície de interacção, que captura e envia uma sequência de vídeo com o estado da superfície a um computador responsável pelo processamento das imagens capturadas. É composto ainda por um projector, também conectado ao computador, responsável por projectar na superfície interactiva a representação visual do estado interno da aplicação. A arquitectura do Reactable (figura 2.6) encontra-se dividida em quatro componentes lógicas que comunicam entre si. Uma componente construída no *framework* *reactIVision* que permite detectar rapidamente o toque de múltiplos dedos em simultâneo e detectar e reconhecer objectos físicos a partir da sequência de vídeo capturada. Esta componente envia mensagens TUIO com a codificação dos eventos detectados ao gestor de conexões (*connection manager*), responsável pela gestão do estado dos objectos detectados na superfície e pela construção da rede de ligações que estes formam no momento. O gestor de conexões é ainda responsável por comunicar aos sintetizadores de áudio e vídeo o estado actual da superfície para que estes possam transformar essa informação em *feedback* auditivo e visual. O sintetizador de áudio recorre a bibliotecas de amostras de áudio e técnicas de síntese e processamento de som

para criar o som desejado. O sintetizador de vídeo, por sua vez, é responsável por processar e enviar ao projector a informação visual a ser transmitida aos utilizadores.

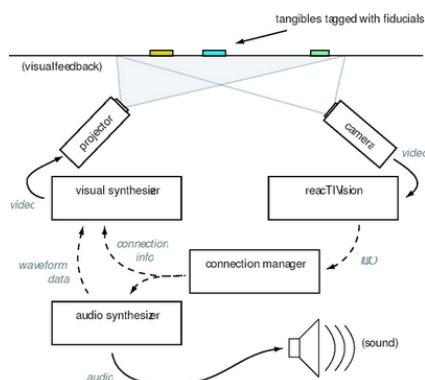


Figura 2.6: Arquitectura Reactable

Actualmente, existem duas versões comerciais do Reactable, o Reactable Live! e o Reactable Experience. O primeiro é o mais recente, uma versão menor, mais portátil, desenvolvida para músicos profissionais. O segundo é o mais parecido com o Reactable original, adequado a espaços públicos e instituições como museus, escolas e universidades. Recentemente também foi lançada uma nova versão, o Reactable mobile, para dispositivos móveis multitoque, como o iPad.

2.1.3 Memtable

A Memtable [Hun09] é uma mesa interactiva multitoque de elevada dimensão, desenvolvida por Seth Hunter no Massachusetts Institute of Technology (MIT), em 2009, no âmbito da sua dissertação de mestrado. A Memtable (figura 2.7) oferece suporte a reuniões de grupo através da captura e armazenamento de todas as interacções físicas e digitais detectadas pela superfície interactiva. Desta forma, todas as interacções que ocorrem numa reunião ficam armazenadas na Memtable e podem ser acedidas mais tarde.

O objectivo da Memtable é capturar o conteúdo, implícito e explícito, resultante das discussões entre participantes de uma reunião, e organizar esse conteúdo para pesquisa e navegação em reuniões posteriores. A Memtable utiliza o potencial de uma superfície multitoque, para permitir a grupos de trabalho de 4 a 8 pessoas, simultaneamente, registar, discutir e recordar informação relevante para as suas reuniões.

Ao contrário da grande maioria das aplicações para mesas interactivas, que se restringem apenas a uma utilização de reduzida duração, a Memtable têm em consideração as implicações de uma interacção a longo prazo, construindo ao longo do tempo uma história de dados, resultando no aumento progressivo da sua utilidade.

Na Memtable cada participante tem acesso a um menu individual que pode ser reposicionado em qualquer local da mesa, a qualquer momento da reunião. Este menu é constituído por uma série de funções de *input* e *output* para criação ou recolha de conteúdo. Durante a reunião os participantes têm acesso a um cronograma para recordar pontos importantes de reuniões



Figura 2.7: A Memtable

passadas, podendo trazer esses pontos para a discussão, arrastando-os para a área de trabalho. Existem quatro teclados para que cada participante possa introduzir anotações de texto a qualquer momento. Os participantes também podem desenhar num dos três blocos de notas disponíveis e transferir uma representação digital desse esboço para a mesa. Objectos físicos e papéis de grandes dimensões podem ser capturados por uma câmara de elevada resolução e enviados para a superfície. A qualquer momento é possível transferir informação de um computador portátil para a área de trabalho. É feita uma gravação contínua de áudio da reunião que fica disponível para reprodução posterior. Finalmente, através de uma caneta, é possível desenhar directamente na superfície da mesa.

A Memtable é constituída por duas câmaras sensíveis a luz infravermelha, dois espelhos, dois projectores e um computador (figura 2.7). Apresenta uma construção original e inovadora, com novos materiais e um *design* que simplifica e melhora esta tecnologia, permitindo melhorar a interacção do grupo de trabalho através de um aumento da superfície interactiva, suportando vários participantes, graças a uma altura mais natural para interacções sociais e finalmente devido a um assento mais ergonómico. Graças ao posicionamento centralizado do equipamento as laterais da mesa são abertas, deixando espaço para as pernas e joelhos.

2.2 Interfaces caligráficas

Interfaces caligráficas ou interfaces pen-based [Mey95] são interfaces que permitem ao utilizador interagir com informação digital, normalmente num Tablet PC, através de uma caneta (*stylus*).

Este tipo de interface tira partido de uma das capacidades mais desenvolvidas pelos seres humanos: escrever numa folha de papel. O facto de o utilizador estar familiarizado com esta metáfora facilita o processo de aprendizagem e garante, desde logo, uma experiência interactiva mais natural para o utilizador.

Curiosamente, o uso de canetas como forma de interacção antecede o aparecimento do próprio rato, surgindo na altura como a escolha óbvia para interagir com um computador. Contudo, devido à incapacidade dos computadores reconhecerem o mais simples dos gestos, muito menos a complexidade da nossa escrita, levou a que a indústria comesse a procurar novos dispositivos, pensados de raiz para a metáfora *desktop*, acabando o rato e o teclado por

tornarem-se os dispositivos de eleição.

Apesar do rato e da caneta serem essencialmente dispositivos apontadores, existem algumas diferenças que é preciso considerar quando se desenvolvem interfaces para estes dispositivos. Por exemplo, com o rato o utilizador aponta de forma relativa, controlando um cursor no ecrã. Com a caneta o utilizador aponta de forma absoluta, sendo necessário tocar com a caneta no local exacto onde se deseja apontar. Esta é uma diferença substancial na medida em que introduz problemas relacionados com a inadequação da caneta aos métodos de interacção criados a pensar no rato. Por exemplo, uma função muito comum do rato é o duplo clique. Devido à natureza da caneta é muito complicado acertar na mesma posição duas vezes consecutivas, tornando esta função inviável para a caneta. Outra diferença substancial é o facto de a interacção com rato ser normalmente acompanhada por um teclado e com caneta ser normalmente feita sem o auxílio de outros dispositivos. Torna-se assim imperativo para o sucesso das interfaces caligráficas encontrar soluções para o *input* de texto.

Existem duas técnicas diferentes para substituir um teclado usando apenas um Tablet PC e uma caneta: é possível usar a caneta para apontar e seleccionar os caracteres num teclado virtual, método muito demorado e pouco prático, ou então aproveitar a capacidade de escrita do utilizador e reconhecer a sua caligrafia para *input* directo dos caracteres. Os traços feitos com a caneta são analisados por *software* de reconhecimento de escrita, que reconhece as formas como sendo de caracteres manuscritos. É feito o mapeamento para os seus representantes virtuais e estes são dados como *input*, como se de um teclado se tratasse. O objectivo deste tipo de *software* é maximizar a taxa de reconhecimento, sabendo contudo que atingir uma taxa de 100% de precisão é impossível, visto que nem o ser humano é capaz de reconhecer todos os manuscritos com exactidão. Um estudo levado a cabo por LaLomia [LaL94] indica que uma taxa igual ou superior a 97% é aceitável para a maioria dos utilizadores.

Dentro do *software* para reconhecimento de escrita podem-se distinguir duas categorias principais.

A primeira usa o método OCR (Optical Character Recognition), já estabelecido em aplicações comerciais (e.g., leitores de faxes), que permite o reconhecimento de símbolos escritos num papel através do reconhecimento e extracção de padrões. É extremamente susceptível a rotações e distorções e é limitado a um número reduzido de *fonts* e tamanhos. Este método é também conhecido por Off-line Recognition pois no momento da análise o texto já é conhecido integralmente.

Por sua vez, o segundo método, denominado Handwriting Recognition, acontece enquanto o utilizador está a introduzir texto. Este método, também conhecido por On-line Recognition, analisa pequenas porções de informação à medida que o utilizador vai escrevendo e apresenta imediatamente os resultados. Esta resposta em tempo-real é uma das principais características destes sistemas, sendo importante notar que o reconhecimento de padrões usado em OCR não é adequado neste caso, visto que o estilo de escrita varia demasiado de pessoa para pessoa.

Apesar de terem sido usadas e desenvolvidas várias técnicas ao longo dos últimos anos (e.g., Hidden Markov Models e redes neuronais [Mey95]) para procurar aumentar a taxa de sucesso dos sistemas de Handwriting Recognition, ainda não se estabeleceu qual a melhor

técnica e os resultados continuam a ser imperfeitos. Contudo, já existem sistemas no mercado com uma elevada taxa de sucesso e tudo aponta para que estes sistemas venham a ser cada vez mais usados no futuro, sobretudo graças ao renovado interesse por este tipo de tecnologia no mercado crescente dos dispositivos móveis.

Para além de disponibilizar uma alternativa válida à ausência de um teclado, um boa interface caligráfica deve também permitir que o utilizador escreva onde desejar e o que desejar. Este tipo de interface não deve restringir onde o utilizador pode escrever e deve permitir que este utilize qualquer carácter especial usado pelas línguas mais comuns, recorrendo a caracteres Unicode e não ao limitativo ASCII. Outro aspecto importante para o sucesso de uma interface caligráfica é permitir ao utilizador que escreva e desenhe sem ter de alternar entre modos, devendo o *software* de reconhecimento ser capaz de distinguir entre estes dois tipos diferentes de *input*. Sempre que o utilizador desenha algo no ecrã o sistema tenta perceber se o utilizador está a desenhar ou a escrever. Caso esteja a escrever, é usado o *software* de reconhecimento de escrita, caso não seja detectada escrita, é usado um *software* dedicado exclusivamente ao reconhecimento de gestos e traços. Esta separação permite criar um novo mecanismos de interação, através da interpretação de determinadas formas, não como escrita, mas sim como comandos especiais. Por exemplo, uma espiral é normalmente associada à operação de apagar. Sistemas mais recentes já utilizam técnicas para detectar mais do que uma caneta (normalmente um dedo), permitindo multitoque para detecção deste tipo de gestos e garantindo uma maior gama de comandos especiais ao utilizador.

Estes mecanismos de interação, aliados ao facto de a interação com caneta permitir maior expressividade, permitem que este tipo de interface seja o ideal para aplicações de desenho.

Seguem-se alguns exemplos dessas aplicações.

2.2.1 Teddy

Teddy [IMT99] é uma interface caligráfica que permite desenhar modelos tridimensionais de forma rápida e simples, através da análise de desenhos 2D criados pelo utilizador. O sistema constrói automaticamente a partir de traços desenhados no ecrã as superfícies poligonais correspondentes, construindo iterativamente o modelo tridimensional desejado.

A principal motivação para este projecto nasceu da necessidade de desenvolver uma ferramenta que permitisse a criação de modelos tridimensionais simples de forma rápida, para auxiliar, por exemplo, nas fases iniciais de projectos que necessitem de uma ferramenta de prototipagem rápida, ou para fins educacionais e recreativos. Apesar de todo o progresso que se tem verificado na área dos sistemas de modelação 3D, estes sistemas continuam a ser difíceis de usar e demasiado complexos para permitir uma criação rápida de modelos tridimensionais. Com o Teddy o utilizador não precisa de se preocupar com manipulação de pontos de controlo ou criação de malhas poligonais, apenas tem de desenhar os traços da silhueta do objecto e o sistema constrói automaticamente a superfície poligonal correspondente. Esta técnica permite que utilizadores sem qualquer experiência em modelação 3D criem modelos tridimensionais expressivos em minutos, graças à simplicidade do sistema e ao facto de a interface assentar

numa metáfora extremamente familiar ao utilizador (figura 2.8). A interface do Teddy, apesar de se basear em dispositivos de entrada 2D (e.g., caneta ou *stylus*), não recorre ao tradicional estilo de interacção WIMP (Window, Icon, Menu, Pointing device) para operações de modelação, como a maioria dos sistemas de modelação 3D. Neste sistema o utilizador define as operações a realizar desenhando traços no ecrã, o sistema deduz a intenção do utilizador e executa as operações, aproveitando desta forma ao máximo as capacidades do dispositivo de entrada.



Figura 2.8: Teddy

As operações disponíveis para a criação de modelos tridimensionais são: criação de novo objecto, desenho, extrusão, corte e suavização. Estas foram implementadas para permitir uma aprendizagem incremental. Os utilizadores podem criar uma série de modelos apenas com a primeira operação (criação) e podem mais tarde expandir incrementalmente o seu vocabulário aprendendo a trabalhar com as restantes.

O processo de criação começa com o utilizador a desenhar a silhueta (linha fechada) do objecto. O sistema constrói automaticamente a malha poligonal do modelo com base na silhueta 2D introduzida. É fundamental que o ponto inicial e final da linha estejam ligados e que a linha não se intersecte a si própria. Assim que um modelo é criado pode ser observado de vários ângulos e o utilizador pode modificá-lo através de várias operações. Este tipo de operações, como já foi referido, depende exclusivamente dos traçados desenhados pelo utilizador no ecrã, não existindo quaisquer botões ou menus para as definir. Se o utilizador desenhar uma linha dentro da silhueta do objecto, esta é projectada para a sua superfície, desta forma o utilizador pode desenhar detalhes directamente no modelo. O utilizador pode apagar rabiscando sobre as linhas, como se estivesse a riscar algo numa folha de papel. Se a linha desenhada dentro da silhueta do objecto for fechada, o sistema entra em modo de extrusão. Neste modo é possível definir alterações à silhueta do modelo, desenhando novas formas adjacentes ao modelo já criado. Durante este modo é também possível suavizar a superfície do modelo, isto é feito rabiscando sobre a zona desejada, à semelhança de quando se deseja apagar. Finalmente, uma linha que intersecte totalmente o modelo é considerada um corte, desaparecendo a metade esquerda e permanecendo a direita. Este conjunto simples de operações permite a utilizadores sem qualquer experiência em modelação, explorar e criar modelos tridimensionais (figura 2.8) de forma rápida e simples.

O Teddy é um bom exemplo de uma aplicação pensada de raiz para as propriedades das interfaces caligráficas e adaptada aos seus dispositivos de entrada, ignorando a norma WIMP

e aproveitando-se da familiaridade do utilizador com a metáfora da caneta e do papel para facilitar o processo de modelação 3D.

A principal falha apontada a este sistema é a ausência de opções de pintura dos modelos. Produzida também por Igarashi, o Chameleon [IC01] surgiu para colmatar essa lacuna e servir de complemento ao Teddy. O Chameleon é uma ferramenta assente nos mesmos princípios do Teddy mas direccionada exclusivamente para a pintura de modelos. Mais tarde surgiria uma versão combinada dos dois sistemas, o SmoothTeddy [IH03], o sucessor de Teddy e Chameleon.

2.2.2 CrossY

CrossY [AG04] é uma aplicação de desenho criada para demonstrar a aplicabilidade do conceito de *crossing-based interface* em aplicações para Tablet PCs, como forma de minimizar parte das dificuldades resultantes da inadaptação das interfaces caligráficas aos princípios estabelecidos para as tradicionais interfaces gráficas do utilizador.

Crossing-based interfaces são interfaces gráficas que usam gestos ou traços como forma de interação, em vez do tradicional conceito de apontar e clicar. Nestas interfaces o utilizador define as operações a realizar desenhando traços no ecrã e não através de cliques do rato (figura 2.9).

A crescente proliferação de computadores e dispositivos móveis com canetas como dispositivo de interação demonstrou que, apesar de extremamente eficiente, o estilo de interação WIMP não é facilmente adaptável a uma interação com caneta. Como já foi mencionado, muitas interações WIMP, originalmente desenvolvidas para o rato, são difíceis de executar com uma caneta num Tablet PC. Para além do exemplo paradigmático do duplo clique, mencionado anteriormente, existem uma série de dificuldades que é preciso considerar, como a oclusão de áreas do ecrã pela mão do utilizador, acesso limitado a atalhos do teclado, dificuldades em utilizar teclas modificadoras (e.g., Shift e Control). Este tipo de dificuldades resulta da diferença substancial que existe entre estes dois tipos de dispositivos e sobretudo devido ao facto da interação com caneta encorajar a uma forma fluida e contínua de interação, enquanto interfaces gráficas desenhadas para ratos insistem em compartimentar as interações numa serie de passos em que o utilizador aponta e clica para desencadear uma acção.

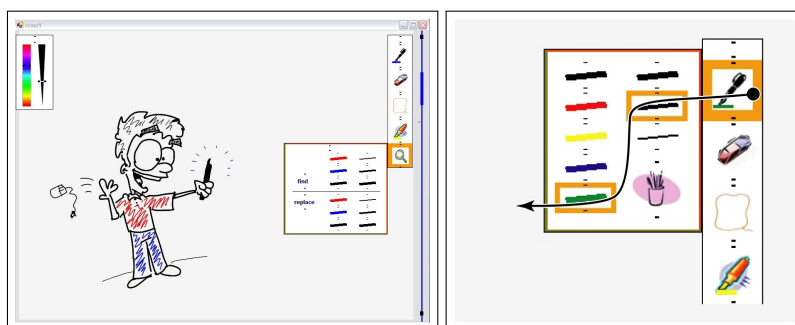


Figura 2.9: CrossY

O CrossY (figura 2.9) foi desenvolvido para demonstrar que a solução para estas dificuldades não reside na adaptação de conceitos previamente estabelecidos, mas sim num novo paradigma de interacção, pensado e adaptado às propriedades do dispositivo em questão, a caneta, e assente nos princípios inovadores das *crossing-based interfaces*.

No CrossY toda a interacção é feita através do reconhecimento de gestos e traços desenhados pelo utilizador no ecrã (figura 2.9), desde a selecção de comandos, até à navegação dentro do documento.

Com o CrossY conseguiu-se demonstrar que é possível criar interfaces expressivas e eficientes para Tablet PCs, que aproveitam ao máximo as capacidades da caneta, desde que sejam pensadas de raiz para o dispositivo em questão e não se recorra a adaptações dos tradicionais conceitos usados pelo rato e teclado. Neste caso recorreu-se ao novo paradigma de *crossing-based interfaces*, que ganha bastante popularidade, sobretudo em dispositivos móveis (e.g., Smartphones e PDAs). Surgem cada vez mais interfaces deste género nestes dispositivos. Um bom exemplo é o popular teclado virtual Swype¹, que recorre a gestos contínuos introduzidos pelo utilizador, quer seja com o dedo, quer seja com uma caneta, de forma a acelerar a introdução de texto.

2.3 Sistemas para a exploração de manuscritos iluminados

Museus e bibliotecas encontram-se entre as instituições que mais investem no desenvolvimento de novas tecnologias de visualização de informação e interfaces de interacção. Procuram sobretudo meios inovadores para transmitirem a sua mensagem e atrair o público mais jovem. Este género de instalações interactivas tem também um papel fundamental para a conservação, restauro [BSX⁺10] e divulgação das obras originais, permitindo que os visitantes tenham acesso às obras sem comprometer a sua integridade. A tendência passa pelo desenvolvimento de instalações interactivas que combinam o virtual e o real de forma a envolver o visitante numa experiência lúdica e enriquecedora.

Estas instalações são desenvolvidas para ensinar através da exploração, deixando uma percepção de entretenimento ao utilizador, sendo este um dos factores que contribui para a aceitação e popularidade que estes sistemas granjeiam. Num estudo recente Horn et al. [HSCJ09, HSJ08] comparou o uso de interfaces tangíveis e interfaces gráficas tradicionais de forma a apurar os benefícios resultantes da aplicação das interfaces tangíveis para fins educacionais. Os resultados demonstram que estes sistemas, quando desenvolvidos cuidadosamente, oferecem de facto um benefício concreto nesse aspecto. Isto acontece sobretudo junto do público mais jovem, com as crianças a demonstrarem maior interesse e a responderem melhor às experiências lúdicas que estes sistemas oferecem.

Dentro do contexto deste trabalho, seguem-se alguns exemplos destes sistemas dedicados à exploração de manuscritos iluminados.

¹<http://swypeinc.com/>

2.3.1 Rerum Novarum

Rerum Novarum: Interactive Exploration of Illuminated Manuscripts [z311, BGC10a] é um projecto desenvolvido pelo Imagelab da Universidade de Módena e Reggio Emília, com o apoio da Biblioteca Estense Universitaria de Módena, para a exploração interactiva de manuscritos iluminados em bibliotecas digitais. O projecto centra-se na concepção e implementação de uma nova geração de sistemas multimédia para análise, pesquisa, recuperação e navegação virtual de imagens e texto em códices iluminados com interfaces naturais. O sistema oferece uma nova experiência de navegação e exploração de obras-primas do Renascimento a especialistas, visitantes de museus e amantes da arte em geral.

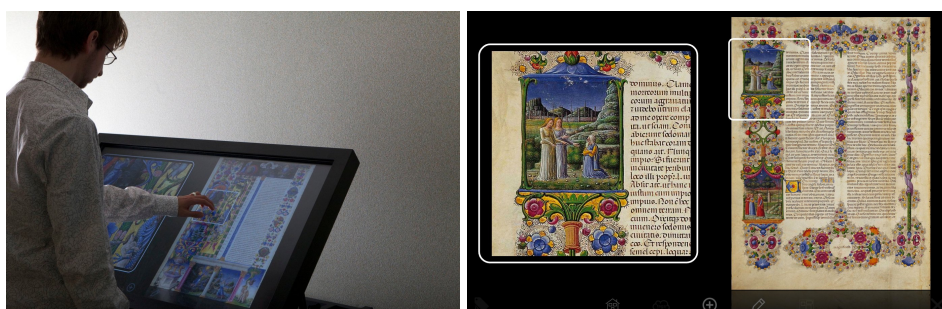


Figura 2.10: Rerum Novarum

Desenvolvido para uma superfície multitoque (figura 2.10), o Rerum Novarum inclui todas interações normalmente disponíveis em sistemas multimédia. Permite explorar a biblioteca digital (útil para quando o utilizador não tem um objectivo definido), permite navegar rapidamente na colecção de manuscritos (útil quando é conhecido um objectivo) e permite ainda procurar rapidamente por manuscritos ou imagens, através de pesquisa textual, semelhanças visuais e *tags* [BGC10b]. O sistema de *tags* é construído à medida que a aplicação é usada, com o *feedback* do utilizador a ter um papel fundamental para enriquecer esta função. Todas as funcionalidades de navegação e pesquisa são construídas em torno de um componente lógico designado *tag navigator*. Este componente fornece acesso às *tags* disponíveis no sistema, permitindo navegar na sua hierarquia e filtrando automaticamente a informação de forma a fornecer o subconjunto exacto de dados que o utilizador deseja consultar.

Todas estas funcionalidades são apresentadas numa interface multitoque (figura 2.10), visualmente apelativa, com suporte para múltiplos utilizadores em simultâneo e com as funcionalidades mais complexas representadas por ícones facilmente distinguíveis, enquanto as funcionalidades mais simples (e.g., selecção, navegação, *zooming*) são geralmente desencadeadas por gestos (e.g., *swipe*, *pinch*).

2.3.2 Turning the Pages™

Turning the Pages™[z411] é uma aplicação desenvolvida pela British Library e pela Armadillo New Media Communications que oferece a bibliotecas e museus uma plataforma de suporte à digitalização de colecções inteiras de livros e manuscritos, ficando disponíveis num ambiente

tridimensional rico e com ferramentas úteis para ajudar na sua compreensão e investigação.

A aplicação oferece um motor gráfico bastante detalhado para representar realisticamente livros e manuscritos em 3D, permitindo que estes sejam examinados de qualquer ângulo e folheados normalmente. A aplicação oferece ainda um conjunto de ferramentas para explorar as obras em bastante profundidade, permitindo, entre outras, *zooming*, pesquisa, anotações pessoais e em grupo.

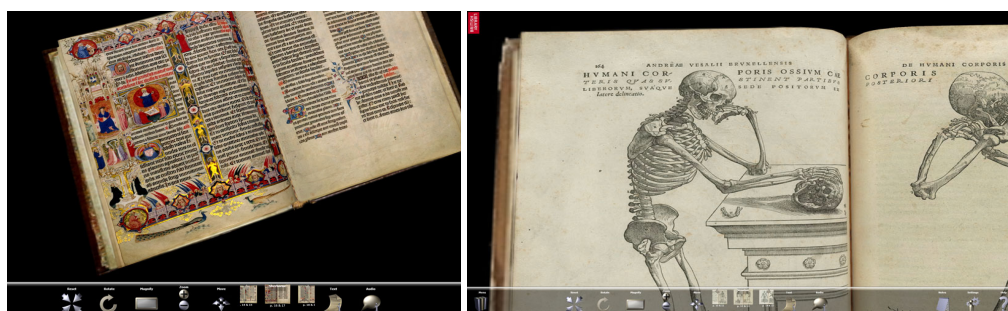


Figura 2.11: Turning the Pages™

O resultado é uma aplicação que apela a diferentes públicos, incluindo profissionais e académicos que procuram acesso às fontes originais para trabalhar, graças sobretudo à componente tridimensional apresentada e à interacção através de uma superfície multitoque. Esta solução permite a consulta e estudo das obras representadas, contribuindo, simultaneamente, para a preservação das obras originais. A aplicação foi desenvolvida para ser executada em quiosques digitais com suporte multitoque, garantindo uma experiência natural e fluida aos seus utilizadores.

2.3.3 Aplicação Web dedicada à produção de manuscritos iluminados

Esta aplicação Web [TMO11], desenvolvida em Flash, encontra-se alojada no site do museu Fitzwilliam, o museu de arte e antiguidades da Universidade de Cambridge, e permite observar o processo criativo de um manuscrito iluminado, desde a confecção do pergaminho, até ao processo de escrita e iluminação.

A interacção com a aplicação é feita com o rato, seguindo instruções textuais que são fornecidas a cada momento e clicando em pontos de interesse que surgem na interface de modo a desencadear determinadas animações representativas dos processos que se pretendem divulgar. A estrutura da aplicação segue precisamente o processo produtivo dos manuscritos iluminados, encontrando-se organizada sequencialmente em vários passos, com as transições a serem desencadeadas pelos utilizadores quando clicam nos pontos de interesse sugeridos. É possível ainda navegar directamente para as várias fases que compõem o processo criativo, clicando em botões presentes ao longo do topo da interface.

O processo de produção dos manuscritos iluminados começa com a confecção dos pergaminhos, o que inclui a recolha, a transformação e a preparação dos materiais necessários à sua

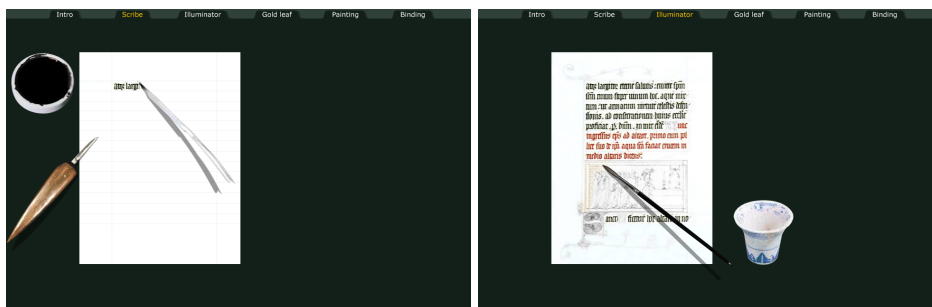


Figura 2.12: Escrita e decoração de manuscritos

confeção. De seguida observa-se a escrita dos textos nos manuscritos (figura 2.12) e o enriquecimento dos textos produzidos com elementos decorativos e ilustrações complementares (figura 2.12). Por fim, observa-se a processo de produção dos códices a partir de um elevado número de fólios.

2.4 Considerações finais

Neste capítulo foram apresentados os conceitos e os aspectos mais relevantes associados aos dois paradigmas de interacção abordados na presente dissertação, assim como as técnicas necessárias para o desenvolvimento da solução proposta.

Na secção 2.1 foram apresentadas algumas motivações para a utilização de sistemas multitoque, assim como soluções de *hardware* e *software* para o desenvolvimento destes sistemas. O estudo destas soluções permitiu avaliar que tipo de configuração poderia ser utilizado na construção da aplicação proposta e perceber que, independentemente da configuração escolhida, todas as soluções estudadas seguem o mesmo padrão arquitectural, assente num modelo cliente-servidor que pressupõe uma separação entre o componente responsável pela detecção de eventos multitoque numa superfície interactiva (servidor) e a camada applicacional (cliente). Esta separação permite que a aplicação seja desenvolvida de forma independente, sem grandes restrições, obrigando apenas à implementação de um cliente TUIO para receber e processar as mensagens emitidas pelo componente servidor. Assim, a aplicação poderá ser futuramente implementada recorrendo a qualquer configuração que suporte o protocolo TUIO, sem perder funcionalidades, facilitando a sua implementação num maior número de locais e, consequentemente, contribuindo para a disseminação da informação veiculada. Foram ainda apresentados alguns exemplos concretos de sistemas multitoque baseados em diferentes combinações de *hardware* e *software*, aplicados a diferentes contextos e a apresentarem diferentes objectivos e funcionalidades. O estudo destes sistemas permitiu sobretudo retirar algumas ilações relativamente ao funcionamento das diferentes configurações usadas, estudar os diferentes tipos de interacção suportados e ainda avaliar como os eventos multitoque influenciam globalmente o funcionamento das aplicações e suas interfaces.

A secção 2.2 centrou-se sobretudo na apresentação de algumas características inerentes às interfaces caligráficas que é preciso considerar quando se desenvolve uma aplicação para um

paradigma consideravelmente diferente daquele que estamos habituados a usar diariamente. Este estudo permitiu perceber algumas das limitações e vantagens que este paradigma introduz e sobretudo como explorar o seu potencial. Foram ainda apresentados alguns exemplos de aplicações de desenho pensadas de raiz como interfaces caligráficas, que tentam tirar partido das características das canetas como meio de interacção e abandonam completamente os princípios normalmente associados às tradicionais *graphical user interfaces*.

Por fim, foram apresentados exemplos da adaptação destas técnicas ao tema dos manuscritos iluminados, o que permitiu avaliar o funcionamento e o desenho das interfaces de algumas aplicações similares às incluídas na instalação interactiva proposta, quer do ponto de vista do tratamento dos conteúdos, quer do ponto de vista das funcionalidades apresentadas.



Exploração de iluminuras medievais

Neste capítulo apresentam-se as aplicações desenvolvidas no âmbito desta dissertação. Para cada uma das aplicações são apresentados em detalhe os conceitos considerados relevantes, as tecnologias utilizadas no seu desenvolvimento, assim como as suas funcionalidades e modos de interacção.

Primeiro é apresentada a aplicação com interface caligráfica, pensada para oferecer aos utilizadores uma experiência de produção de iluminuras aproximada à do *workshop*, e depois a aplicação com interface multitoque, destinada a uma superfície interactiva de grandes dimensões, que tem como objectivo reunir num só local toda a informação disponível sobre o tema.

3.1 Scriptorium Virtual

Um dos pontos mais importantes do estudo levado a cabo pelo Departamento de Conservação e Restauro (DCR) e que é essencial transmitir de forma adequada, diz respeito à criação de iluminuras e todo o processo envolvido na sua produção, desde a mistura dos materiais durante a preparação das tintas, até ao processo de pintura propriamente dito.

Com o objectivo de transmitir todos os aspectos envolvidos na produção de iluminuras no período medieval, o DCR organiza periodicamente um *workshop* onde disponibiliza as ferramentas e os ensinamentos para que os participantes possam aprender e experimentar tudo o que está envolvido nesse processo e possam criar as suas próprias iluminuras. Os participantes do *workshop* são encaminhados passo-a-passo por todo o processo envolvido na reprodução de uma iluminura, através da consulta de um guia que os informa a cada momento sobre os procedimentos a seguir para a iluminura escolhida.

Apesar destes procedimentos variarem ligeiramente consoante a iluminura escolhida, geralmente os passos a seguir são similares. Em primeiro lugar é necessário copiar o desenho

original para que este possa ser reproduzido. Colocando uma folha de papel vegetal sobre a iluminura, o participante copia todos os contornos do desenho original para a folha de papel vegetal. Terminada a cópia, é colocada a folha de papel vegetal sobre o pergaminho e o participante volta a desenhar todos os contornos do desenho, desta vez no verso do papel vegetal, para que todos os traços do desenho original fiquem marcados no pergaminho.

Reproduzidos os traços do desenho original no pergaminho, segue-se o próximo passo, que consiste em passar os contornos a tinta ferrogálica. Esta tinta, que apresenta uma cor negra em tons de roxo, é feita de sais de ferro e ácido tânico a partir de fontes vegetais, e foi o padrão, quer ao nível da escrita, quer ao nível do desenho a tinta, na Europa, desde o século XII até ao século XIX, permanecendo em uso até ao século XX. No *workshop* esta tinta é aplicada no pergaminho com um aparo, que é o instrumento mais parecido com as penas de aves usadas na altura.

Depois de passado o desenho a tinta ferrogálica, segue-se a preparação das tintas. Para cada uma das iluminuras existem tintas que é preciso produzir, todas elas a partir de materiais próprios e com processos de preparação ligeiramente diferentes. Geralmente os processos de preparação envolvem a colocação de um mineral ou pigmento num almofariz, a sua moagem com um pilão e a adição de um ligante, como água ou cola de pergaminho, até atingir-se a consistência e a tonalidade desejada.

Finalizada a preparação das tintas, os participantes devem colorir o pergaminho desenhado anteriormente com as tintas acabadas de produzir, usando pincéis e consultando cuidadosamente uma versão completa do desenho de forma a colorir correctamente a iluminura. Esta fase de pintura envolve ainda uma fase de enriquecimento do desenho através da pintura de pormenores a branco para dar mais expressividade à iluminura criada. Finalmente, os participantes reproduzem o texto original, completando assim a iluminura.

Infelizmente, por mais eficaz e informativo que o *workshop* seja, este tipo de eventos são sempre complicados de organizar, sobretudo em termos logísticos, ficando a periodicidade com que é realizado bastante aquém da desejada e limitando bastante o número de participantes. Outro aspecto com alguma relevância é o facto de o *workshop* incluir algumas tintas que na sua composição, abrangem alguns materiais considerados tóxicos. Visto que os participantes são encorajados a preparar as tintas que posteriormente vão usar, isto implica o manuseamento de matérias perigosas por parte de pessoas sem qualquer tipo de experiência nesta área, o que acarreta um risco bastante elevado. Este aspecto torna-se ainda mais preocupante se considerarmos que o público-alvo do *workshop* é normalmente o público mais jovem.

Surgiu assim a ideia de criar uma aplicação recorrendo às mais recentes técnicas de interacção computacional, com o objectivo de virtualizar todos os aspectos e procedimentos envolvidos no *workshop*, mitigando assim grande parte das dificuldades organizativas apontadas anteriormente. Uma aplicação que ofereça uma experiência atractiva e lúdica, respondendo às expectativas tecnológicas do público mais jovem e permitindo, graças à sua portabilidade, chegar a um maior número de pessoas e alcançar novos públicos em novos locais, maximizando assim a divulgação da informação disponível.

A aplicação poderá ser implementada isoladamente ou conjuntamente com as restantes

aplicações que compõem a instalação interactiva, em locais de interesse cultural, como as instituições onde as obras são preservadas (e.g., bibliotecas, museus, mosteiros). A aplicação poderá ainda ser utilizada como extensão do *workshop*, oferecendo uma nova abordagem com as suas próprias especificações.

Para cumprir os objectivos apontados e poder ser implementado numa situação de utilização real, o sistema deve preencher uma série de requisitos. A aplicação deve incluir todos os aspectos envolvidos no *workshop*. Isto significa não só a divulgação da informação disponível, como também a reprodução de todos os processos manuais de forma ajustada e aproximada à real. O sistema deve permitir que o utilizador produza as suas próprias iluminuras tal como são produzidas no *workshop*, desenhando, preparando tintas e colorindo, acompanhando-o sempre passo-a-passo com informações relevantes.

Assentando a aplicação essencialmente em processos que implicam desenhar e sendo esta actividade bastante rica e expressiva, a opção mais natural e adequada para este tipo de funcionalidades foi desenvolver a aplicação para tablets. Recorrer a canetas como dispositivo de entrada, para além de oferecer um método de interacção mais natural e ser a escolha óbvia para uma aplicação com estas características, permite também usufruir dos mais recentes avanços nesta área, com canetas activas capazes de detectar pressão e inclinação, condições ideais para capturar a expressividade inerente a esta actividade. Pretende-se também com o recurso a este tipo de dispositivo, contribuir para uma maior abstracção do dispositivo físico usado, imergindo ao máximo o utilizador na experiência do *workshop* virtual. Contribui também para esta imersão o *design* da aplicação, que é adequado à temática retratada.

Como principal requisito, para além da divulgação explícita de informação relacionada com as iluminuras, como a sua produção e o seu contexto histórico e social, é importante também transmitir alguns aspectos relacionados com a produção das iluminuras de forma implícita, ou seja, exclusivamente através da interacção do utilizador com a aplicação. Um destes aspectos é capturar parte das dificuldades subjacentes à produção de iluminuras, encontradas no *workshop* e ainda mais proeminentes no *scriptorium* medieval, de modo a recriá-las na aplicação. Contrariamente aos procedimentos habituais no desenvolvimento de *software*, isto é, tornar a aplicação o mais acessível possível para o utilizador, neste caso existe uma preocupação em manter até certo ponto algumas das dificuldades dos processos que se pretendem simular. Dificuldades presentes sobretudo durante a fase de desenho e pintura, com o utilizador, por exemplo, a ver-se impossibilitado de apagar, visto que não existia essa possibilidade na altura, ou a ser obrigado a molhar regularmente o instrumento de pintura no recipiente da tinta durante a fase de desenho, como acontece na realidade. É importante reproduzir estes aspectos para transmitir aos utilizadores, mesmo que de forma menos óbvia, as dificuldades existentes na época, neste caso experimentando essas mesmas dificuldades durante o processo de criação.

Apesar de tudo, é importante mencionar que estes aspectos foram pensados apenas para a fase de desenho e pintura, não influenciando em nada o desenho da interface, tendo sido esta pensada para ser o mais acessível possível ao utilizador. A aplicação deve proporcionar aos utilizadores uma interacção agradável e informativa com o tema das iluminuras medievais,

deve ser de fácil aprendizagem e fácil utilização para os diferentes tipos de utilizadores, visto que, apesar de ser destinado ao público mais jovem, a aplicação será utilizada por utilizadores de todas as idades e de diferentes níveis de experiência.

3.1.1 Produção de iluminuras

Na aplicação proposta, a produção de iluminuras é feita essencialmente através do acompanhamento de um tutorial, constituído por um conjunto de passos extraídos directamente do *workshop* real. Começa com o utilizador a escolher a iluminura a produzir, seguindo-se os passos inerentes à produção da iluminura seleccionada. O utilizador pode ainda consultar informação complementar sobre a iluminura ou experimentar a preparação das tintas incluídas na produção da iluminura escolhida.

Cada uma das iluminuras é representada no sistema por um conjunto de elementos multimédia. Este conjunto contém toda a informação disponível sobre a iluminura seleccionada, isto é, informação relativa aos passos imprescindíveis para a sua reprodução, as tintas usadas, informações sobre a sua concepção e o seu contexto histórico e social, e ainda os métodos usados na preparação das tintas, o que inclui técnicas, materiais e instrumentos. Estes elementos multimédia, que são maioritariamente imagens, apesar de vídeo também poder ser utilizado, constituem uma forte componente da aplicação, desempenhando um papel importante no desenho da interface.

As funcionalidades da aplicação encontram-se distribuídas por diferentes janelas, com estas a preencherem totalmente o ecrã do tablet e as suas transições a serem desencadeadas pelos utilizadores. As janelas da aplicação, assim como as funcionalidades presentes em cada uma, são apresentadas de seguida.

Seleção de iluminura: Permite ao utilizador consultar as iluminuras disponíveis no sistema e seleccionar aquela que pretende produzir.

Produção de iluminura: Permite ao utilizador desenhar e colorir todos os pormenores relativos à iluminura escolhida, sempre com o auxílio de um tutorial composto por um conjunto de passos extraídos directamente do *workshop* real. Durante a produção, o utilizador tem acesso às tintas indispensáveis à reprodução da iluminura escolhida, e apesar de instruído sobre qual a tinta a usar em cada passo do tutorial, pode escolher a que desejar a qualquer momento. Visto que a tinta seleccionada é consumida durante a sua utilização, o utilizador é forçado a seleccionar regularmente a tinta que deseja usar, de modo a refrescar a tinta usada na caneta. O utilizador tem ainda acesso a um conjunto de instrumentos de desenho e pintura, necessários para a produção da iluminura escolhida, que são simulados através da parametrização automática das definições da caneta para cada passo do tutorial.

Consultar informação sobre iluminura: Permite ao utilizador consultar informação global e navegar num conjunto de informações detalhadas sobre a iluminura que se encontra no momento a produzir.

Consultar detalhes sobre iluminura: Permite ao utilizador explorar alguns aspectos em maior detalhe, respeitantes à iluminura escolhida.

Preparação de tintas: Permite ao utilizador experimentar todo o processo envolvido na preparação das tintas usadas na produção da iluminura escolhida, auxiliados por um tutorial composto por um conjunto de passos inspirados no *workshop* real. Nesta janela o utilizador tem acesso a todos os instrumentos, materiais e técnicas, necessários para a preparação da tinta escolhida.

Com estas funcionalidades encontram-se reunidas as condições para que a aplicação cumpra com os objectivos e requisitos definidos anteriormente.

3.1.2 Interacção

A interacção com a aplicação proposta é feita através de uma interface caligráfica, com os utilizadores a interagir com a aplicação num tablet, recorrendo a uma caneta. Este tipo de interface, para além de constituir a escolha mais natural para esta aplicação, tendo em conta que um dos seus objectivos é simular um processo que na sua essência é fundamentalmente desenhar, permite ainda, como já foi referido, usufruir dos mais recentes avanços na área. Com canetas capazes de detectar muitos níveis de pressão, o utilizador sente-se a trabalhar com os tradicionais instrumentos de pintura, abstraído do facto de estar na realidade a trabalhar com um dispositivo electrónico. Este tipo de tecnologia permite ao utilizador, por exemplo, ajustar dinamicamente a espessura da linha e a quantidade de tinta usada, simplesmente variando a pressão da caneta sobre a superfície do tablet. Assim, torna-se desnecessária a presença de menus e botões, permitindo uma interacção muito mais rica e próxima da experiência do *workshop*.

A interacção com a aplicação é feita somente com a caneta, sem o auxílio de qualquer outro dispositivo. A ausência de outros dispositivos de entrada, sobretudo a ausência de um dispositivo com as características de um teclado ou mesmo de um rato, foi um factor importante no desenho da interface da aplicação. A interface foi desenhada, não só para se ajustar e tirar o máximo partido das características do dispositivo de entrada usado, mas também para atenuar a ausência de outros dispositivos.

Foi dada especial atenção a um problema inerente às interfaces caligráficas, o facto de o utilizador ter de apontar de forma absoluta, tocando em todos os locais desejados com a ponta da caneta. Isto acarreta uma dificuldade extra para a interacção, demorando o utilizador mais tempo a executar cada acção. Este problema, identificado em 2.2 e apontado como uma dificuldade de interacção resultante da diferença substancial que existe entre a interacção feita com um rato e com uma caneta, introduz mais algumas dificuldades que é necessário considerar na fase de desenho, como a oclusão de áreas da interface pela mão do utilizador e dificuldades em apontar com a caneta e executar operações mais complexas, como o duplo clique.

Para tentar minimizar parte destas dificuldades, as funcionalidades da aplicação foram organizadas por áreas bem identificadas e compartimentadas, com as funcionalidades mais relevantes a ocuparem zonas de maior destaque, sem correrem o risco de serem obstruídas pelas

mãos e braços dos utilizadores, e com as acções similares e complementares juntas, de forma a minimizar o desgaste e o tempo despendido no seu acesso durante uma utilização normal.

A navegação é feita sempre através de botões bem identificados e de dimensões consideráveis, com as acções a serem desencadeadas com um único toque da ponta da caneta na superfície interactiva do tablet e com a área de desenho e pintura bem identificada e destacada.

Como já foi referido, a interface encontra-se organizada em janelas, com as principais funcionalidades da aplicação a ter uma janela dedicada a preencher totalmente o ecrã do tablet. A interface inclui uma janela para a selecção de iluminuras, outra janela dedicada à produção de iluminuras, uma janela para a preparação das tintas, uma janela com informações sobre a iluminura escolhida e finalmente uma com detalhes relevantes sobre o tema. As transições entre janelas são despoletadas pelos utilizadores com a caneta, pressionando botões representativos das funcionalidades pretendidas.

O *design* da aplicação é inspirado na temática retratada, com os instrumentos, os materiais e alguns elementos decorativos presentes nas iluminuras a constituírem a maior inspiração. Com isto pretende-se transmitir alguma informação sobre o assunto aos utilizadores, capturando parte do ambiente do *workshop* e do *scriptorium* medieval, na aplicação desenvolvida.

O desenho inicial da interface, proposto pelos responsáveis do DCR, sofreu uma série de alterações ao longo da fase de desenvolvimento. Estas alterações serviram fundamentalmente para aperfeiçoar a navegação na interface e para uma melhor adaptação desta às características inerentes ao estilo de interacção usado, de forma a melhorar a experiência do utilizador com a aplicação. Foram fundamentais para estas alterações os resultados obtidos em diversas reuniões entre os vários elementos que participam no projecto e as conclusões retiradas de uma avaliação da interface levada a cabo por alguns peritos do DCR, numa fase adiantada do desenvolvimento, que pode ser consultada na secção [5.1.1](#).

3.1.2.1 Colecção de iluminuras medievais

Nesta janela são apresentadas todas as iluminuras disponíveis no sistema para reprodução (figura [3.1](#)). Cada uma é representada por uma imagem circular composta pelo seu nome e por uma pré-visualização da iluminura retratada. Estes *thumbnails* encontram-se espalhados por toda a área disponível da interface e têm uma ordenação e posicionamento inicial predefinido. Este posicionamento é determinado com base no número de iluminuras disponíveis no sistema, o aproveitamento do espaço disponível e a importância de cada uma das iluminuras representadas, com as mais relevantes a gozarem de maior destaque em posições centrais da interface. Apesar das posições iniciais predefinidas, os *thumbnails* são susceptíveis de serem arrastados e largados (*drag-and-drop*) pelos utilizadores em qualquer posição da interface que desejarem, permanecendo estes nesse mesmo local. Para seleccionar uma iluminura basta tocar com a caneta no respectivo *thumbnail*, sendo o utilizador imediatamente encaminhado para a janela de produção de iluminuras.



Figura 3.1: Janela de selecção de iluminuras

3.1.2.2 Produção de iluminuras

Após ter escolhido a iluminura que pretende reproduzir, o utilizador é encaminhado para esta janela. Aqui o utilizador tem à sua disposição os instrumentos e as instruções necessárias para reproduzir da melhor forma possível a iluminura escolhida. A interface (figura 3.2) é composta por três áreas principais: a área de desenho, as tintas disponíveis e o tutorial da iluminura escolhida.

Claramente destacada e ocupando a maior parte do espaço disponível na interface, encontra-se a área de desenho. Nesta área o utilizador tem acesso a todos os materiais e instrumentos de desenho usados no *workshop* e pode desenhar, colorir e escrever, seguindo sempre os passos do tutorial, de forma a reproduzir a iluminura seleccionada.

A simulação dos instrumentos de pintura é feita automaticamente de acordo com os passos do tutorial. Para cada passo do tutorial os parâmetros associados à caneta são ajustados automaticamente. Assim, o utilizador tem acesso a todos os instrumentos necessários (e.g., canetas de aparo e pincéis de diferentes tamanhos e feitios) sem precisar de se preocupar com menus. Para desenhar com qualquer um destes instrumentos de pintura, basta o utilizador tocar e arrastar a ponta da caneta na região destinada a essa finalidade. Graças à capacidade da caneta em detectar a pressão exercida sobre a superfície do tablet, é possível ao utilizador controlar com precisão a quantidade de tinta usada e a espessura do traço desenhado, mais uma vez sem recorrer a menus. Para além disso, como já foi referido anteriormente, de forma a recriar a utilização de instrumentos reais, a tinta seleccionada é consumida durante a sua utilização, sendo necessário refrescar a selecção com alguma regularidade. Estas funcionalidades permitem capturar parte da complexidade e expressividade subjacente à actividade de desenhar, contribuindo para a produção de iluminuras mais ricas, com maior detalhe e mais próximas das reais (figura 3.3).



Figura 3.2: Janela de produção de iluminuras

São ainda exibidas imagens auxiliares na área de desenho, que assistem o utilizador durante o processo de reprodução à medida que este vai avançando no tutorial. Apesar de este processo variar consoante a iluminura escolhida, normalmente começa com exibição de uma imagem com os contornos da iluminura, permitindo ao utilizador desenhar por cima desta, simulando o uso de papel vegetal no *workshop*. De seguida, é exibido o desenho criado anteriormente sobre a iluminura original, para que o utilizador possa observar as cores e o local onde estas devem ser aplicadas. Finalmente, surge o desenho criado pelo utilizador com fundo de pergaminho, pronto a ser colorido (figura 3.4). A partir deste momento são passos específicos de cada uma das iluminuras, relativos à forma com devem ser pintadas e enriquecidas com texto.

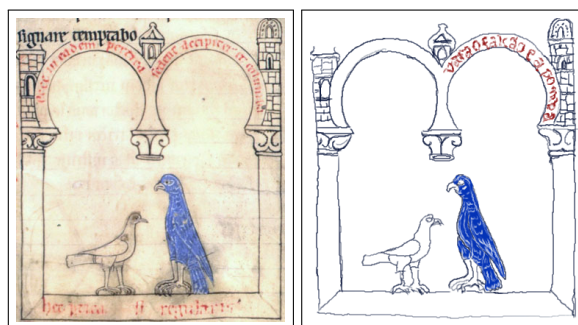


Figura 3.3: Iluminura original vs. iluminura digital

Localizadas à direita da área de desenho, ocupando menos espaço, encontram-se as tintas disponíveis para a iluminura escolhida. Para cada iluminura disponível no sistema, existe um conjunto de tintas necessárias à sua reprodução, que são carregadas automaticamente quando



Figura 3.4: Utilizador a pintar iluminura

da escolha da iluminura pretendida e ficam disponíveis durante toda a fase de desenho e pintura.

As tintas são representadas por imagens de conchas decoradas com as respectivas cores (figura 3.2). Para seleccionar qualquer uma das tintas o utilizador deve pressionar com a caneta a respectiva concha, reconstituindo assim o procedimento original de mergulhar, por exemplo, a pena no recipiente da tinta. Junto a cada uma das conchas é visível um ícone com um pequeno almofariz. Ao pressionar este botão o utilizador é encaminhado para a janela de preparação da respectiva tinta, podendo voltar a qualquer momento e retomar o trabalho exactamente do ponto onde se encontrava.

No canto superior esquerdo da interface encontra-se o botão que permite desencadear a transição para a janela informativa (figura 3.6). Aqui o utilizador pode consultar informações sobre a iluminura seleccionada, explorar o significado dos vários elementos que a compõem, perceber a sua importância e o seu contexto histórico e social. Sempre que o utilizador navega para esta janela, é mantido o estado da iluminura na área de produção. Desta forma o utilizador pode consultar a janela informativa a qualquer momento sem perder o trabalho realizado.

No canto superior direito da interface encontra-se o botão que permite retroceder para a colecção de iluminuras (figura 3.1). Quando isto sucede, é dado como terminado o processo de produção e é armazenada automaticamente a iluminura criada no respectivo repositório. Caso o tablet se encontre ligado a uma impressora, é possível ainda imprimir a iluminura criada, para que os utilizadores possam levar uma recordação consigo. Terminado o processo de criação, o utilizador poderá escolher qualquer uma das iluminuras disponíveis para reprodução, reiniciando assim o processo.

Localizado no fundo da interface, por baixo da área de desenho e tintas, encontra-se o tutorial. Aqui o utilizador tem acesso a informações detalhadas passo-a-passo sobre o processo de reprodução da iluminura escolhida.

O tutorial (figura 3.2) é composto por um conjunto de passos, que individualmente representam uma tarefa extraída directamente do *workshop* real, o que abarca todas as instruções e instrumentos necessários à sua conclusão, nomeadamente: a designação do instrumento de pintura necessário para esse passo, o que permite configurar dinamicamente os parâmetros

associados à caneta e evitar assim o recurso desnecessário a menus de selecção; a imagem auxiliar que surge na área de desenho e que permite ao utilizador, por exemplo, desenhar por cima como se de papel vegetal se tratasse; uma imagem com informação textual e que permite ao utilizador saber exactamente o que fazer; finalmente uma imagem auxiliar que permite complementar a informação textual, quer com representações iconográficas do passo em questão, quer com visualizações da iluminura original para consulta de detalhes.

Assim que o utilizador escolhe a iluminura e é encaminhado para a respectiva janela de produção, é logo confrontado com o primeiro passo do tutorial, que é carregado automaticamente. Normalmente neste passo inicial são oferecidas breves instruções ao utilizador sobre o funcionamento global da aplicação, o que esta oferece e alguns esclarecimentos em relação à iluminura seleccionada. Nos passos seguintes surge a produção da iluminura propriamente dita, que apesar de variar consoante a iluminura escolhida, começa normalmente pela fase de desenho dos traços da iluminura a partir da original, seguindo-se a fase de pintura e o enriquecimento da iluminura com elementos caligráficos. Esta mudança de passos é desencadeada pelo utilizador, podendo este retroceder ou avançar a qualquer momento, simplesmente pressionando os botões presentes do lado esquerdo e direito do tutorial. Todas as mudanças de passo implicam o carregamento automático de todos os elementos associados ao passo em questão.

Para além dos botões de progresso, destacam-se duas áreas no tutorial. Uma área com instruções, presente no lado direito, que permite a cada momento exibir imagens com informação textual para cada um dos passos, e uma área com informação auxiliar, presente no lado esquerdo, que permite complementar a informação textual através da exibição de imagens que possam ser relevantes para a conclusão de cada um dos passos. Esta última, como já foi referido anteriormente, tem como principal objectivo a exibição da iluminura original para consulta de detalhes. O principal obstáculo encontrado foi o reduzido espaço disponível para exibir tanta informação, neste caso, imagens de alta resolução com todos os detalhes associados a estas obras.

Uma solução possível seria adaptar a imagem à dimensão da janela e adicionar botões de *zoom* ou exibir a imagem na sua dimensão original e adicionar *scrollbars* para navegação. Contudo, esta solução é demasiado complexa para o tipo de interface e usabilidade que se pretende para a aplicação. O recurso a botões de reduzidas dimensões, não só aumenta o número de passos necessários para executar uma simples operação, como introduz dificuldades, graças à especificidade do dispositivo de entrada usado. Tocar com a caneta em botões de dimensões reduzidas pode revelar-se uma tarefa extremamente complexa, sobretudo para pessoas com pouca experiência com este tipo de dispositivos. Para além disso, a adição de botões ou *scrollbars* iria consumir parte substancial do espaço disponível e não se adaptaria à estética pretendida para a aplicação. A solução passou por uma abordagem mais acessível, tirando partido das características do dispositivo de entrada usado, ou seja, exibe-se a imagem na sua dimensão original, permitindo ao utilizador navegar e explorar os detalhes presentes na imagem, arrastando-a com a caneta. Assim, é possível ao utilizador explorar os detalhes da iluminura rapidamente sem recorrer a quaisquer botões. Para efeitos de usabilidade foi ainda adicionado

kinetic scrolling, que consiste no deslocamento de uma longa lista com um pouco de física envolvida, neste caso aplicado ao deslocamento de uma imagem de alta resolução num espaço confinado. O utilizador ao arrastar e soltar a imagem, permite que esta continue em movimento com a mesma velocidade, desacelerando depois com o tempo até parar completamente. Um efeito similar ao produzido quando se gira uma roda. Tornado popular no iPhone, este efeito rapidamente invadiu outras plataformas móveis com ecrãs tácteis, devido à sua utilidade face ao método convencional das *scrollbars* e sobretudo por se adaptar perfeitamente ao método de interacção usado.

3.1.2.3 Preparação de tintas

Junto a cada uma das tintas na janela de produção de iluminuras, encontra-se um botão que permite desencadear a transição para a janela de preparação da respectiva tinta. Nesta janela o utilizador tem acesso aos instrumentos, aos materiais e às instruções necessárias para preparar a tinta seleccionada.

Tal como na produção de iluminuras, o conceito adoptado para o processo de preparação de tintas é directamente inspirado nos métodos usados no *workshop*. Não obstante este processo variar conforme a tinta escolhida, geralmente os processos de preparação envolvem, ou a adição de materiais para dentro de um recipiente preparado para o efeito, normalmente um almofariz, ou a moagem e mistura do seu conteúdo com um instrumento adequado, geralmente um pilão.

A aplicação desenvolvida simula estes processos, representando todos os materiais e instrumentos como imagens, exibindo-as de forma organizada na interface e permitindo que estas sejam manuseadas pelo utilizador, recriando assim a manipulação real dos materiais no *workshop*. A manipulação dos diferentes materiais é executada com a caneta, através do arrastamento das respectivas imagens para uma área onde se encontra uma representação do estado actual do almofariz. A manipulação dos materiais engloba ainda a capacidade de misturar e moer o conteúdo do almofariz virtual, simulando a utilização de um pilão real com a caneta. Durante este processo a medição da distância percorrida pela caneta, assim como a capacidade de detectar a pressão exercida por esta na superfície do tablet, permitem ao utilizador controlar a velocidade de progressão do nível de moagem e mistura com maior precisão.

A interface (figura 3.5) apresenta uma estrutura similar à usada na janela de produção de iluminuras, sendo composta também por três áreas principais: a área de mistura onde se encontra o almofariz, os materiais necessários e o tutorial de preparação da tinta.

Localizada no lado direito da interface, à direita da área de mistura, encontra-se a área onde são expostos os materiais. Aqui o utilizador tem acesso a todos os materiais necessários à preparação da tinta seleccionada, bastando para isso, como já foi mencionado, arrastar com a caneta o material desejado para dentro do almofariz. Alguns materiais permanecem indisponíveis após a sua utilização, outros materiais, como a água ou cola de pergaminho, estão sempre disponíveis para o utilizador poder adicionar a qualquer momento a quantidade que desejar.

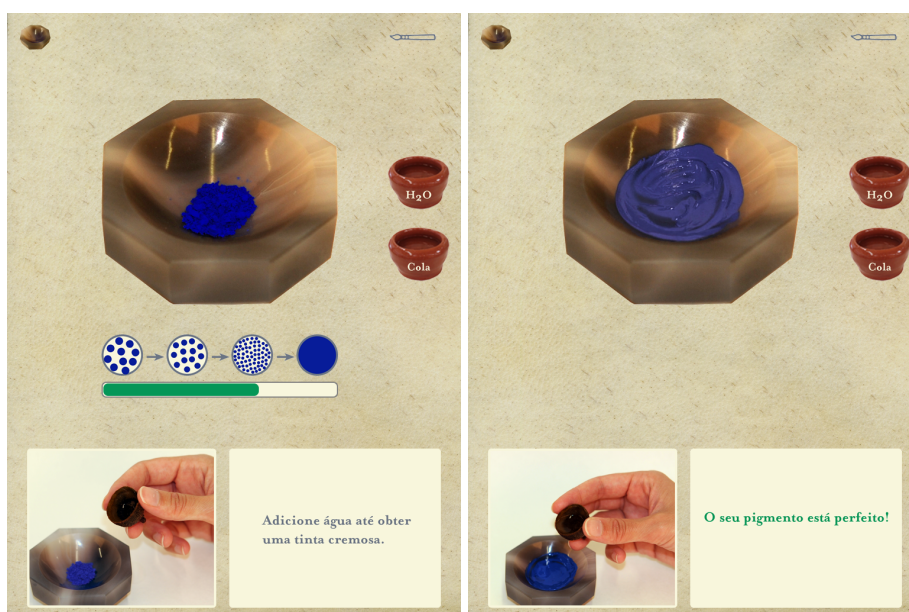


Figura 3.5: Janela de preparação de tintas

A área de mistura encontra-se claramente destacada e ocupa a maior parte do espaço disponível nesta janela. Nesta área é exibido o almofariz, que vai sofrendo alterações à medida que os utilizadores vão adicionando, misturando e moendo novos materiais. As interações que o utilizador tem com esta área são essencialmente de dois tipos, nomeadamente, a adição de novos materiais através do seu arrastamento desde a área onde são exibidos até ao local do almofariz, e a moagem e mistura dos materiais que já se encontram no seu interior. O processo de preparação das tintas consiste essencialmente numa combinação destes dois tipos de interações, sendo esta combinação diferente para cada uma das tintas disponíveis no sistema.

O tutorial consiste fundamentalmente numa receita composta por uma sequência de passos que o utilizador deve seguir de forma a preparar a tinta seleccionada. A cada novo passo do tutorial corresponde uma actualização do almofariz, criando assim a ilusão de uma evolução constante da tinta preparada e um *feedback* visual das acções realizadas pelo utilizador. Estão ainda incluídas em cada passo do tutorial duas imagens, que à semelhança do que acontece durante a produção de iluminuras, permitem informar o utilizador sobre o que deve fazer a cada momento. Estas imagens são exibidas no fundo da interface, por baixo do almofariz e dos materiais.

Contrariamente ao que sucede no tutorial de produção de iluminuras, onde as mudanças de passos são despoletadas explicitamente pelo utilizador, neste caso as transições são desencadeadas implicitamente, ou seja, a mudança de passos e correspondente actualização do estado do almofariz e instruções é despoletada automaticamente pelas acções praticadas pelo utilizador, dispensando completamente a presença de botões de progresso. Como consequência, esta mudança de passos apresenta uma estrutura bastante rígida, não permitindo recuar e avançando apenas quando é detectado um evento que respeite as condições necessárias à preparação da tinta. Estas condições estão associadas aos dois tipos de eventos que podem ocorrer e podem

estar presentes individualmente ou em simultâneo no mesmo passo do tutorial. No caso de um evento de adição de novo material ao almofariz, a transição só é executada se esse passo contemplar um evento desse tipo que inclua o material adicionado. No caso de um evento de moagem ou mistura do conteúdo do almofariz, a transição só é desencadeada quando se atingir o nível de moagem estabelecido para esse passo. Este nível pode ser acompanhado pelo utilizador, quer através do *feedback* visual dado pelo almofariz, quer através da consulta de dois componentes que surgem quando o utilizador inicia o processo de mistura e moagem. É exibida uma imagem representativa da evolução da moagem dos minerais e uma barra de progresso, que, conjuntamente, informam o utilizador sobre o estado actual do composto presente no almofariz.

Para além da informação adquirida através da observação directa do estado da barra de progresso, é possível ainda perceber a evolução e o estado do processo de moagem através da observação das diferentes cores que esta barra adquire ao longo do tempo. A cor vermelha indica que o composto ainda não está pronto e que precisa de ser moído para poder adquirir a consistência e tonalidade desejada. A cor verde, por sua vez, indica que o composto está pronto, podendo o utilizador proceder à adição do próximo material indicado e prosseguir com a preparação da tinta. Em alguns casos pode acontecer o utilizador ignorar, quer o que lhe é pedido para fazer nas instruções, quer o *feedback* visual do almofariz e da barra de progresso, e continuar a moer mesmo depois de ter atingido a cor verde na barra de progresso. Nestes casos o composto acaba por ultrapassar a tonalidade desejada, ficando demasiado deteriorado para poder ser usado, restando apenas uma solução, reiniciar o processo. Presente no canto superior esquerdo da interface, encontra-se um botão que permite ao utilizador reiniciar o processo de preparação da tinta, quando são cometidos erros ou quando, por qualquer razão, o utilizador deseja repetir o processo.

Quando o utilizador é confrontado pela primeira vez com a janela de preparação de tintas, encontra um almofariz vazio, os materiais necessários alinhados ao lado do almofariz e em baixo, as primeiras instruções para começar a preparar a tinta. Geralmente os processos de preparação começam com o arrastamento do mineral ou pigmento indicado para dentro do almofariz, seguindo-se a sua moagem com a caneta até ao nível estabelecido, o arrastamento de um ligante para dentro do almofariz e terminando com a mistura do composto até atingir-se a consistência e tonalidade desejada. Terminada a tinta o utilizador pode voltar à janela de produção de iluminuras tocando no botão presente no canto superior direito.

3.1.2.4 Consultar informação sobre iluminura

Todas as iluminuras disponíveis no sistema incluem um conjunto de informações acessível ao utilizador para consulta e exploração, a partir do botão presente no canto superior esquerdo da interface da janela de produção de iluminuras.

Nesta janela informativa (figura 3.6), o utilizador encontra informações gerais sobre a iluminura, como o seu significado, a sua relevância, o seu contexto histórico, social e político, e ainda uma série de detalhes relativos a alguns elementos que a constituem. A informação

geral é apresentada em maior destaque e ocupa grande parte da área disponível, os detalhes surgem em baixo, organizados numa lista, com cada tópico a ser representado por uma pré-visualização do seu conteúdo. Esta lista, que pode ser constituída por um elevado número de tópicos consoante a iluminura escolhida, apresenta no máximo três pré-visualizações a cada momento, podendo o utilizador aceder às restantes através de dois botões presentes no lado esquerdo e direito da lista para retroceder e avançar, respectivamente. Para consultar em maior detalhe um dos tópicos disponíveis, o utilizador deve pressionar a respectiva pré-visualização, o que desencadeará a transição para uma nova janela com a informação desejada.

Nesta janela (figura 3.6) geralmente são apresentadas imagens e texto para veicular a informação pretendida, com a imagem destacada a ocupar grande parte da área disponível e o texto por baixo a servir de legenda. Para voltar à janela anterior encontra-se um botão directamente por baixo da informação apresentada, que permite desencadear a respectiva transição.

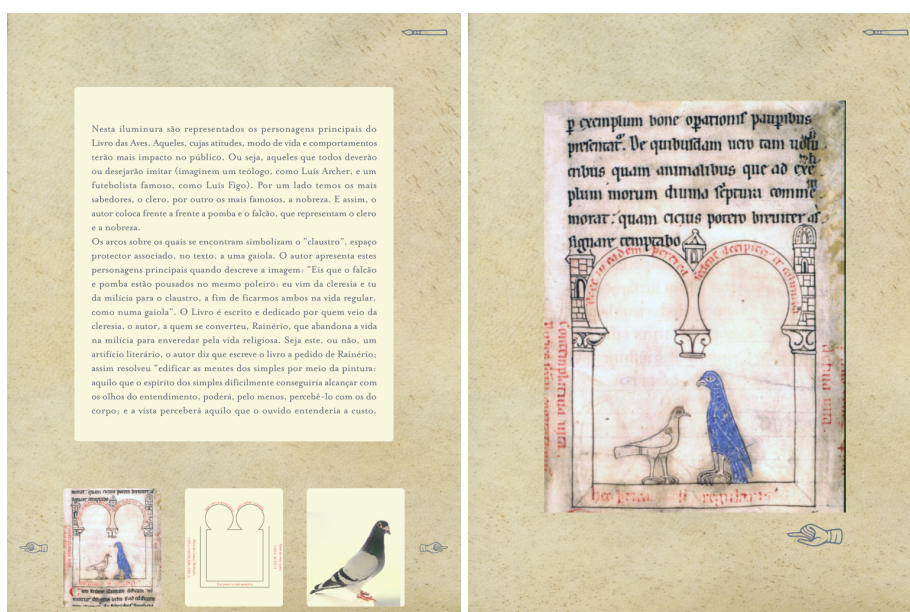


Figura 3.6: Janelas com informações sobre as iluminuras

Em qualquer uma destas janelas informativas encontra-se um botão no canto superior direito da interface que permite a qualquer momento voltar à janela de produção de iluminuras e continuar o trabalho exactamente do mesmo ponto.

3.1.3 Tecnologias utilizadas

A aplicação foi desenvolvida em linguagem C++, recorrendo à plataforma Qt [z511], uma biblioteca *open source* de desenvolvimento multiplataforma em C++, distribuída pela Nokia gratuitamente e que permite o desenvolvimento de *software* com interfaces gráficas.

A organização do conteúdo da aplicação, por sua vez, foi feita de forma estruturada num ficheiro XML.

Em termos de *hardware*, foi usado durante o desenvolvimento da aplicação, assim como



Figura 3.7: HP Compaq tc4200 Tablet PC

nas avaliações realizadas, um HP Compaq tc4200 Tablet PC (figura 3.7) com controladores (*device driver*) da marca Wacom. Apesar de este tablet permitir desenhar e pintar com uma caneta sensível à pressão, encontra-se prevista a implementação da versão final em modelos de tablets mais recentes, que ofereçam ainda mais sensibilidade à pressão e globalmente uma melhor performance.

3.2 Painel interactivo para exploração de iluminuras medievais

Um dos objectivos do projecto, que define o âmbito desta dissertação, é a criação de uma aplicação interactiva para divulgar os resultados do estudo sobre a cor em iluminuras medievais Portuguesas. Uma aplicação que permita divulgar a informação disponível e promover o tema das iluminuras junto do público, mas que também proporcione uma interacção agradável e lúdica aos seus utilizadores.

Para além da divulgação dos resultados do estudo, existe também uma forte preocupação em contextualizar a informação divulgada, integrando-a de forma natural com alguma da informação disponível sobre a Idade Média. Assim, a aplicação prevê não só a divulgação dos resultados do estudo, como também a divulgação do seu contexto histórico, social, político e artístico.

Outro objectivo da aplicação é permitir aos utilizadores explorar algumas das obras mais marcantes e representativas do espólio de iluminuras medievais Portuguesas. A principal motivação é dar aos utilizadores a possibilidade de consultarem obras que em condições normais não estariam ao seu alcance, visto tratarem-se de património histórico com um valor inestimável, só estando disponíveis a especialistas da área com a devida autorização. A exploração digital destas obras permite não só a sua divulgação e estudo, como contribui fortemente para a sua preservação.

Pretende-se ainda que a aplicação reúna num só local todas as iluminuras criadas pelos utilizadores no Scriptorium Virtual e permita a sua consulta por parte dos utilizadores, estabelecendo assim uma ponte entre as duas aplicações e contribuindo para a coesão da instalação interactiva.

Sendo o principal objectivo da aplicação divulgar a informação disponível em locais de interesse cultural, a opção mais natural e adequada a este tipo de funcionalidades foi desenvolver a aplicação para uma superfície interactiva multitoque de grandes dimensões. A proliferação

deste tipo de aplicações em locais de interesse cultural é actualmente uma realidade. Assiste-se cada vez mais à procura deste tipo de soluções por parte de entidades que pretendem divulgar a sua mensagem de forma inovadora, tentando corresponder às expectativas de um público cada vez mais exigente.

O sistema poderá ser implementado isoladamente ou conjuntamente com as restantes aplicações que compõem a instalação interactiva, em locais de interesse cultural, como as instituições onde as obras são preservadas (e.g., bibliotecas, museus, mosteiros), ou em qualquer local que beneficie da presença da aplicação.

Para cumprir os objectivos apontados e poder ser implementada numa situação de utilização real, a aplicação deve preencher um conjunto de requisitos. O principal requisito diz respeito, sobretudo, ao modo como a informação é divulgada. A divulgação da informação deve ser feita de forma clara, organizada e de fácil navegação. A interface da aplicação deve incentivar à exploração da informação e fomentar o interesse do utilizador pelo tema. A aplicação deve proporcionar aos utilizadores uma interacção agradável e informativa, deve ser de fácil aprendizagem e fácil utilização para os diferentes tipos de utilizadores a que se destina.

A aplicação é dirigida a um público heterogéneo, com utilizadores de todas as idades e diferentes níveis de experiência. Esta diversidade de utilizadores exige que a aplicação seja o mais abrangente possível, para que qualquer que seja o utilizador, este possa ganhar algo com a utilização da aplicação, quer seja um utilizador experiente em busca de mais informações, quer seja um utilizador que pretende apenas explorar o sistema.

3.2.1 Consulta e exploração de iluminuras medievais

A informação disponibilizada pela aplicação para consulta e exploração é representada no sistema por imagens, ou seja, todos os textos, ilustrações, fotografias e diagramas que constituem a informação semântica divulgada pela aplicação e compõem a sua interface, encontram-se armazenados em ficheiros de imagens. A interface da aplicação encontra-se organizada em diferentes janelas, com estas a preencherem totalmente a área visível da superfície interactiva, a serem dedicadas a um tópico em particular e a apresentar um conjunto específico de funcionalidades. A consulta de informação pode ir desde a simples leitura de textos e visualização de imagens, até à exploração de códices virtuais. As principais janelas da aplicação, assim como as suas funcionalidades, são apresentados de seguida.

Portugal Românico: Janela dedicada à exploração do contexto histórico, social, político e artístico em que as iluminuras foram criadas. Esta informação, apresentada essencialmente sob a forma de texto e imagens, encontra-se organizada e categorizada num conjunto de submenus e janelas para simplificar a sua consulta e exploração.

Mosteiro: Janela simples, constituída essencialmente por texto e imagens, dedicada à divulgação de informação sobre os mosteiros onde eram criadas as iluminuras e sobre o quotidiano dos monges que as criavam.

Santa Cruz, Lorvão e Alcobaça: Janelas simples, constituídas essencialmente por texto e imagens, dedicadas à divulgação de informação relacionada com estes três mosteiros, que tiveram um papel preponderante na criação de alguma das iluminuras mais marcantes do espólio de iluminuras medievais Portuguesas.

As cores: Janela dedicada à divulgação dos resultados do estudo sobre a cor em iluminuras medievais Portuguesas. Constituída por um conjunto de submenus e janelas, que organizam e categorizam a informação de modo a simplificar a sua exploração e consulta. A informação, apresentada essencialmente sob a forma de texto e imagens, diz respeito sobretudo aos modos de preparação, às aplicações, às combinações e aos significados, das várias cores que foram alvo de estudo.

Apocalipse do Lorvão e Livro das Aves: Janelas dedicadas à exploração e consulta de códices virtuais. Aqui o utilizador pode consultar e explorar cada uma destas obras, folheando os seus fólios naturalmente como se de um códice real se tratasse.

Scriptorium: Janela dedicada à divulgação de informação sobre o *scriptorium*, onde eram criadas as iluminuras. Esta janela prevê a exibição de uma ilustração de um *scriptorium* medieval, com um conjunto de pontos de interesse que o utilizador pode explorar de modo a obter informações sobre o respectivo assunto.

Códice Medieval: Janela dedicada à divulgação de informação relacionada com os códices medievais. Esta janela apresenta uma combinação das funcionalidades exibidas pelos códices virtuais e pela janela dedicada ao *scriptorium* medieval, permitindo consultar e explorar os códices virtuais folheando os seus fólios naturalmente, mas desta feita, com uma camada de informação sobreposta aos fólios, com um conjunto de pontos de interesse que o utilizador pode explorar de modo a obter informações sobre determinadas particularidades dos códices explorados.

Galeria: Janela dedicada à consulta e exploração das iluminuras criadas pelos utilizadores na aplicação Scriptorium Virtual.

3.2.2 Interacção

A interacção com a aplicação proposta é feita através de uma interface tangível numa superfície interactiva multitoque de grandes dimensões. Os utilizadores interagem com o sistema através do toque dos dedos na sua superfície interactiva, podendo usar uma ou duas mãos. Este tipo de interacção compreende também a execução de alguns gestos específicos, que permitem realizar determinadas acções em certas áreas da interface. O principal tipo de interacção com o sistema, que diz respeito à navegação na sua estrutura e à exploração da informação divulgada, é feita através do toque do utilizador em áreas bem identificadas e destacadas da interface, nomeadamente, botões e ícones representativos dos tópicos a explorar.

O desenho da interface resultou do trabalho da equipa de projecto e sofreu diversas alterações ao longo do processo de desenvolvimento. As alterações visaram sobretudo o melhoramento organizativo da informação divulgada, a adaptação dos diferentes tipos de informação aos modos de interacção e a simplificação da navegação na interface da aplicação.

O *design* da interface é fortemente inspirado na temática retratada, com alguns elementos decorativos das iluminuras a terem um papel preponderante na definição do estilo visual da aplicação. Com isto pretende-se envolver mais o utilizador na experiência e contribuir para a adaptação do sistema ao local onde será implementado. Mesmo considerando a diversidade informativa e funcional que a aplicação apresenta nas diferentes janelas que a compõem, foi feito um esforço de harmonização da interface, no que concerne aos elementos visuais usados. Um exemplo disso são os botões utilizados na interface, representados sempre por imagens circulares com pré-visualizações da informação que representam.

A navegação na interface da aplicação apresenta-se organizada hierarquicamente, numa estrutura em árvore cuja raiz corresponde ao menu inicial e onde cada nó diz respeito a uma janela dedicada a um tópico em particular. Devido aos utilizadores tendencialmente abandonarem a aplicação sem voltarem ao menu inicial, está prevista uma transição automática para o menu inicial depois de cinco minutos de inactividade.

3.2.2.1 Menu inicial

Nesta janela (figura 3.8) incluem-se todos os temas disponíveis para consulta e exploração. Cada tema é representado por um botão circular, composto pelo seu título e uma pré-visualização da informação que representa. Os temas disponíveis, descritos anteriormente em 3.2.1, são: «Portugal Românico», «Mosteiro», «Santa Cruz», «Lorvão», «Alcobaça», «As cores», «Apocalipse do Lorvão», «Livro das Aves», «Scriptorium», «Códice Medieval» e «Galeria». Esta janela prevê ainda uma rede de ligações entre os botões da interface, que permite ao utilizador aferir visualmente as relações existentes entre os diferentes temas disponíveis, assim como a intensidade dessas mesmas relações.

Para consultar e explorar cada um dos temas apresentados, o utilizador deve pressionar com o dedo o respectivo botão, desencadeado a transição para a janela com a informação desejada. A nova janela tanto pode conter informação sob a forma de texto e imagens, como sucede em algumas janelas simples descritas anteriormente em 3.2.1, como pode ser mais complexa, podendo apresentar uma combinação de submenus e janelas com informação cada vez mais específica, ou apresentar algumas funcionalidades próprias, como sucede nos códices virtuais e na galeria.

Devido à aplicação apresentar uma estrutura de navegação hierarquizada, com vários níveis de profundidade, a sua interface contempla, qualquer que seja a janela em que o utilizador se encontra, a possibilidade de retroceder para a janela anterior ou retroceder para o menu inicial, através de dois botões bem identificados e destacados na interface. Encontra-se também prevista uma transição automática para esta janela após cinco minutos de inactividade, para que os utilizadores possam sempre começar a sua interacção com o sistema do menu inicial.

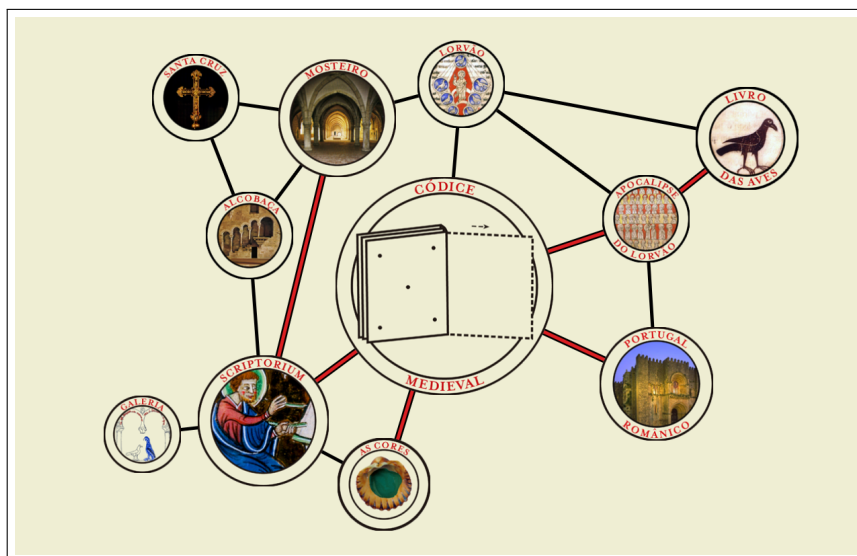


Figura 3.8: Menu inicial

3.2.2.2 Portugal românico

Esta janela é composta por um submenu com os diferentes tópicos disponíveis sobre o tema em questão. Cada tópico, tal como sucede no menu inicial, é representado por um botão circular composto pelo seu título e uma pré-visualização da informação que representa. Os tópicos disponíveis são: «Onde e quando», «Culturas e saberes», «Arte» e «Homens e mulheres». Para consultar e explorar cada um destes tópicos, o utilizador deve pressionar com o dedo o respectivo botão, desencadeando a transição para a janela com a informação desejada.

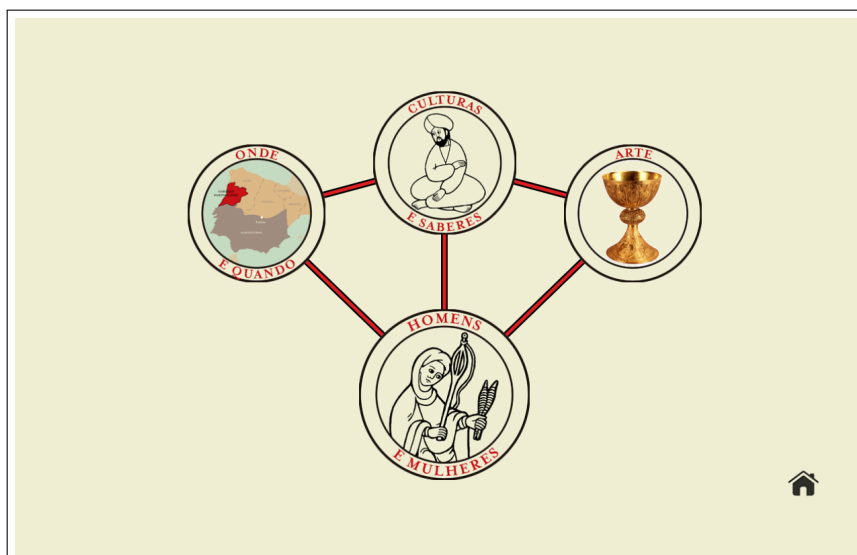


Figura 3.9: Menu «Portugal Românico»

A janela «Onde e quando» prevê a exibição de imagens, texto e algumas animações com informações sobre o contexto histórico e político em que foram criadas as iluminuras.

A janela «Culturas e saberes» prevê a exibição de imagens e texto com informações sobre

o contexto sociocultural em que foram criadas as iluminuras. Esta janela inclui ainda alguns botões que permitem explorar mais a fundo alguns aspectos relacionados com o assunto abordado, no que diz respeito às diferentes culturas que constituíam a sociedade Portuguesa na Idade Média. O utilizador ao pressionar estes botões é conduzido a uma nova janela, com informação relativa à cultura escolhida sob a forma de texto e imagens.

A janela «Arte» prevê a exibição de texto e imagens com informações sobre o contexto artístico em que foram criadas as iluminuras.

Por fim, a janela «Homens e mulheres» prevê a exibição de texto e imagens com informações sobre as diferentes classes sociais e figuras que povoavam a sociedade Portuguesa na Idade Média.

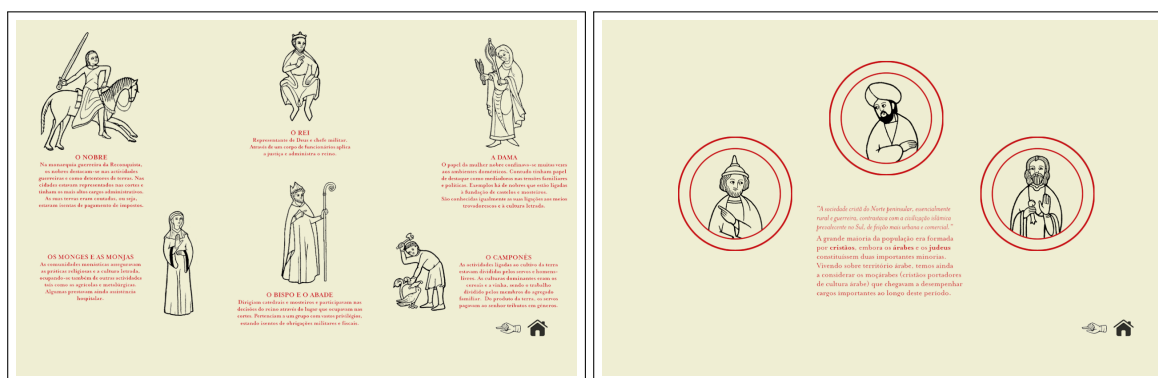


Figura 3.10: Exemplos de janelas de «Portugal Românico»

3.2.2.3 As cores

Esta janela é composta por um submenu com todas as cores disponíveis na aplicação. Cada cor é representada por um botão circular composto pelo seu nome e uma imagem de uma concha com a respectiva cor. Para consultar e explorar cada uma das cores o utilizador deve pressionar com o dedo o respectivo botão, desencadeando a transição para a janela com todos os tópicos sobre a cor escolhida.

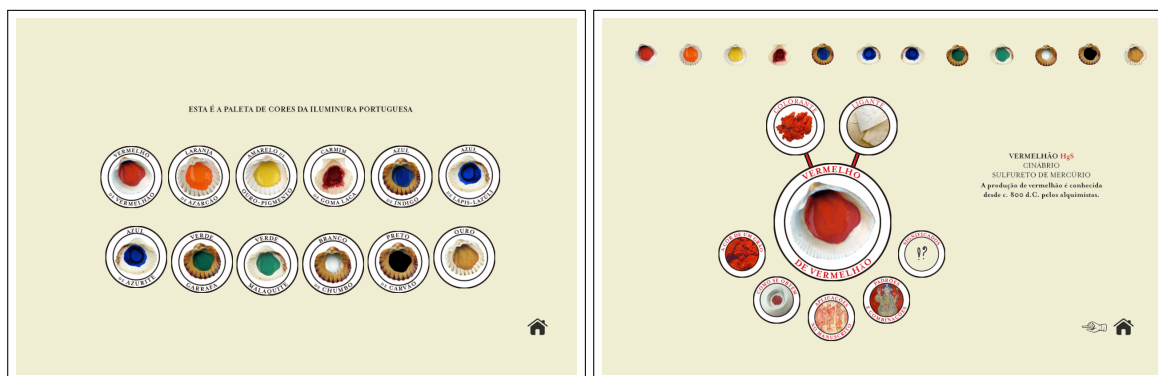


Figura 3.11: Menu de selecção de cor e menu da cor escolhida

Os tópicos disponíveis para cada uma das cores são: «Colorante», «Ligante», «A cor de um grão», «Como se obtém», «Aplicações no manuscrito», «Padrões e combinações» e «Significados».

Estes tópicos, por sua vez, também são representados por botões circulares compostos pelos seus títulos e pré-visualizações da informação que representam. A informação que estes botões representam pode surgir sob a forma de um *pop-up* nessa mesma janela ou implicar a transição para uma nova janela, que por sua vez, poderá também conter um submenu e/ou informação sob a forma de texto, imagens e *pop-ups*.

Sendo a janela «As cores» dedicada precisamente aos resultados do estudo que se pretende divulgar, é natural que este compreenda mais informação que os restantes. Isto implicou um esforço suplementar na categorização e organização da informação e teve como consequência um aumento significativo da profundidade apresentada pela estrutura de navegação. Para facilitar a navegação nesta estrutura e evitar que o utilizador tenha de voltar sempre à janela anterior para escolher uma nova cor, encontram-se disponíveis nesta janela, botões que permitem uma mudança rápida entre as diferentes cores disponibilizadas pelo sistema.



Figura 3.12: Exemplos de janelas das cores

3.2.2.4 Códices virtuais

Nestas janelas (figura 3.13) o utilizador pode consultar e explorar os códices disponíveis na aplicação, nomeadamente, o «Apocalipse do Lorvão» e o «Livro das Aves», folheando os seus fólios como se de códices reais se tratassem. Para folhear o códice o utilizador deve pressionar e arrastar com os dedos as pontas inferiores dos fólios como se estivesse a folhear páginas de um livro. A mudança de fólio é controlada integralmente pelo utilizador e é acompanhada pelo som de um livro a ser folheado. Para além disso, apresenta visualmente uma aproximação ao comportamento real das folhas de um livro, permitindo emergir o utilizador ao máximo na consulta das obras.



Figura 3.13: Exemplo de códice virtual

3.2.2.5 Scriptorium

Esta janela (figura 3.14) é composta por uma ilustração de um *scriptorium* medieval, com um conjunto de pontos de interesse que o utilizador pode explorar. Cada ponto de interesse é representado na ilustração por uma área bem destacada e identificada para o efeito. Para explorar cada um dos pontos de interesse o utilizador deve pressionar com o dedo a respectiva área, desencadeando o surgimento de um *pop-up* nesse local, com a informação desejada.

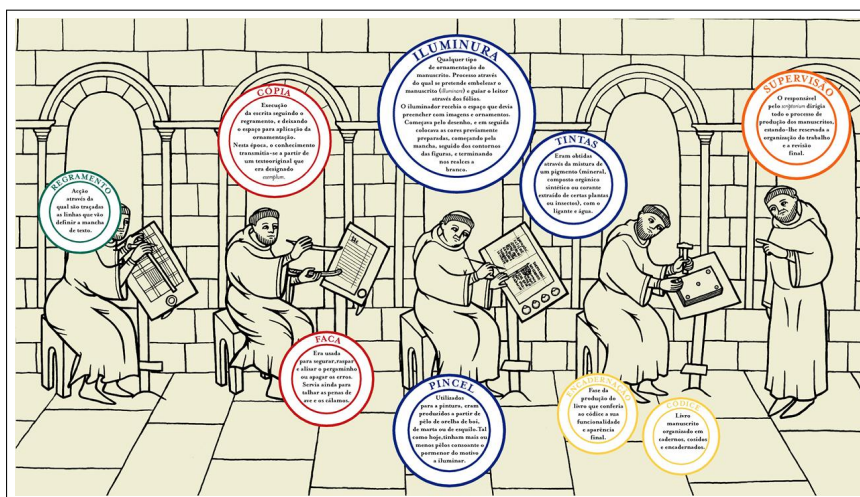


Figura 3.14: Janela «Scriptorium»

3.2.2.6 Códice medieval

Nesta janela (figura 3.15) o utilizador pode consultar e explorar um códice composto por uma selecção de fólios, provenientes de um conjunto alargado de códices que foram alvo de estudo. O critério de selecção baseia-se no facto dos fólios apresentarem algum aspecto que se pretende

destacar e divulgar na aplicação. O utilizador pode folhear os fólhos normalmente, como sucede nas janelas dos códices virtuais, mas desta feita, com uma camada de informação sobreposta aos fólhos, como sucede na janela «Scriptorium». Assim, para além de folhear normalmente os fólhos do códice, o utilizador pode explorar cada um dos pontos de interesse disponíveis, bastando para isso pressionar com os dedos a respectiva área, desencadeando o surgimento de um *pop-up* com a informação desejada.



Figura 3.15: Janela «Códice medieval»

3.2.2.7 Galeria

Nesta janela (figura 3.16) apresentam-se todas as iluminuras criadas pelos utilizadores na aplicação Scriptorium Virtual. As iluminuras surgem dispostas na horizontal, lado a lado, podendo o utilizador navegar na lista pressionado e arrastando com os dedos as iluminuras que são visíveis nesse instante. A interação nesta janela compreende ainda *kinetic scrolling*, que permite uma navegação mais natural e satisfatória aos utilizadores. O *kinetic scrolling* é activado quando o utilizador, após arrastar as iluminuras a determinada velocidade, levanta o dedo da superfície interactiva. Após esse instante, as iluminuras permanecem em movimento durante algum tempo, dependendo a duração desse movimento da velocidade detectada inicialmente. O desenvolvimento do *kinetic scrolling* pode ser consultado em maior detalhe na secção 4.1.5.

Para consultar em detalhe as iluminuras, o utilizador deve pressionar com o dedo a iluminura desejada. Após a selecção, a iluminura surge nas suas dimensões reais, destacada no centro da interface. Para voltar novamente à lista, basta o utilizador tocar novamente na superfície interactiva.

3.2.3 Tecnologias utilizadas

A aplicação foi desenvolvida recorrendo à plataforma openFrameworks [LW11], uma biblioteca *open source* em C++, desenvolvida por Zachary Lieberman e Theo Watson, adequada para

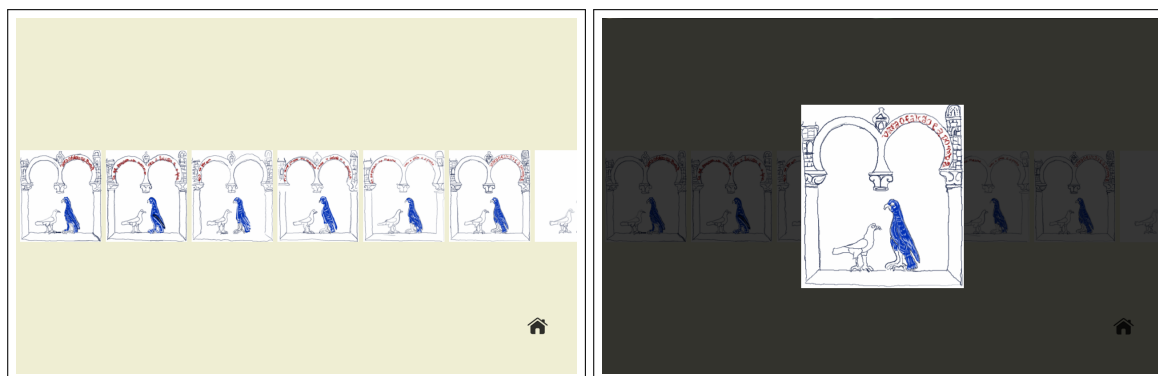


Figura 3.16: Galeria de iluminuras

o desenvolvimento de aplicações criativas, com uma forte componente visual. O openFrameworks permite a criação de interfaces criativas e oferece uma vasta biblioteca de suporte à manipulação de áudio, vídeo e imagem. Para além disso, o openFrameworks suporta *addons*, bibliotecas adicionais desenvolvidas por terceiros, que podem ser adicionados aos projectos para estender a sua funcionalidade.

A organização do conteúdo da aplicação, por sua vez, foi feita de forma estruturada em vários ficheiros XML.

Em termos de *hardware*, a aplicação encontra-se preparada para ser implementada em qualquer tipo de configuração que recorra ao protocolo TUIO para transmitir eventos multitoque detectados numa superfície interactiva.

4

Realização do sistema

Neste capítulo apresentam-se os aspectos considerados mais relevantes da produção da instalação interactiva desenvolvida no âmbito desta dissertação. São ainda apresentadas as dificuldades encontradas durante a realização do sistema e as soluções implementadas como resposta.

4.1 Scriptorium Virtual

O processo de desenvolvimento desta aplicação começou pela produção da estrutura de suporte na biblioteca Qt, seguindo-se o desenvolvimento da interface das janelas que constituem a aplicação, assim como os mecanismos associados a cada uma. Em paralelo foram sendo criadas as estruturas auxiliares para o armazenamento da informação necessária à aplicação, assim como o componente que permite carregar esses dados para memória e disponibilizá-los aos restantes elementos que compõem a aplicação.

4.1.1 Qt

Como foi referido anteriormente em 3.1.3, a aplicação foi desenvolvida em linguagem C++, recorrendo à plataforma Qt [z511], uma biblioteca *open source* de desenvolvimento multiplataforma em C++, distribuída pela Nokia gratuitamente e que permite o desenvolvimento de *software* com interfaces gráficas.

A escolha recaiu sobre esta plataforma pela sua versatilidade e sobretudo pelo suporte que oferece à produção de *software* para tablets. O Qt permite a detecção de eventos gerados por diferentes tipos de dispositivos sobre a superfície de um tablet e a obtenção de uma série de informações relativas a essa interacção. Permite a detecção de eventos gerados por canetas, o dispositivo usado nesta aplicação, por *pucks*, *airbrushes*, ratos 4D e *rotation stylus* (*stylus 6D*).

Possibilita o reconhecimento do tipo de ponta usada por cada um destes dispositivos, distinguindo entre pontas de caneta, borracha e cursores gerados por *pucks*. Permite ainda medir um conjunto de valores a partir da interacção destes dispositivos com a superfície de um tablet, nomeadamente: a posição do dispositivo relativamente à janela da aplicação que recebeu o evento; a pressão exercida pelo dispositivo sobre a superfície do tablet, com os valores a variarem entre 0.0 e 1.0; a rotação do dispositivo em graus, valor normalmente produzido por ratos 4D; e o ângulo entre o dispositivo (e.g., uma caneta) e a superfície do tablet, com os valores a variarem entre -60° e $+60^\circ$.

Visto que estas informações são fornecidas ao Qt pelo controlador do dispositivo (*device driver*), para tirar partido destas funcionalidades é necessário que o *hardware* suporte a detecção e leitura deste tipo de dados e que os controladores usados disponibilizem esta informação às camadas superiores, neste caso ao Qt. Tendo em conta a diversidade de tablets existentes no mercado, as diferentes funcionalidades que estes oferecem e a dependência do Qt em relação às interfaces oferecidas pelos controladores, compreende-se que o Qt tenha como requisito a utilização exclusiva de controladores distribuídos por um único fabricante, neste caso a marca Wacom que é uma referência nesta área. Esta restrição permite ao Qt uniformizar os mecanismos de detecção de eventos e captura de informação e abstrair-se do dispositivo físico usado, tendo como base a interface disponibilizada pelos controladores desenvolvidos por este fabricante, que oferecem diferentes tipos de dados medidos da superfície de um tablet. Estes controladores tanto podem ser encontrados nos dispositivos distribuídos pela Wacom, como podem ser instalados em tablets de outros fabricantes, permitindo assim tirar partido destas funcionalidades em qualquer tablet que cumpra os requisitos. Neste caso, como já foi referido em 3.1.3, foi usado durante a fase de desenvolvimento um HP Compaq tc4200 Tablet PC, com os controladores disponibilizados pela Wacom na sua página na Internet.

Para além deste suporte oferecido pela plataforma à produção de *software* para tablets, fundamental para a aplicação desenvolvida, o Qt oferece ainda um conjunto de funcionalidades e ferramentas que auxiliam o processo de desenvolvimento e que merecem destaque, nomeadamente, APIs em C++ que introduzem novas classes e estendem a linguagem, vasta documentação sobre a plataforma disponível *online*, vasta biblioteca de componentes (*widgets*) para a construção de interfaces gráficas (e.g., botões, menus, janelas, ícones), um *kit* de desenvolvimento de *software* (SDK) que combina a biblioteca Qt com uma série de ferramentas projectadas para simplificar a criação de aplicações com interfaces gráficas, suporte multiplataforma e suporte à internacionalização. Todas estas funcionalidades contribuem para um ambiente propício à produção de aplicações com interfaces gráficas atractivas e funcionais, cenário ideal para a aplicação desenvolvida.

Na base de qualquer aplicação desenvolvida em Qt encontra-se a biblioteca de classes e métodos que estendem a linguagem C++. Classes que permitem, para além do acesso a bases de dados SQL, *parsing* XML, suporte a *threads*, suporte à comunicação em rede, manipulação de ficheiros, reprodução de áudio e vídeo, entre outros, permitem também a representação de uma extensa colecção de componentes gráficos (*widgets*) que constituem o principal instrumento para a construção de interfaces gráficas complexas e atraentes na plataforma Qt.

Na base destes *widgets* e de todos os componentes que formam a extensa biblioteca de recursos gráficos do Qt, encontra-se a classe `QObject`. Esta classe é o coração do modelo de objectos Qt. O modelo de objectos padrão em C++ oferece um suporte em tempo de execução bastante eficiente para o paradigma de objectos, mas a sua natureza estática é demasiado inflexível para determinados domínios. A programação de interfaces gráficas é um domínio que requer, tanto eficiência em tempo de execução, como um elevado nível de flexibilidade. A plataforma Qt oferece isto, combinando a velocidade do C++ com a flexibilidade do modelo de objectos Qt. Os recursos oferecidos por este modelo e pela classe `QObject` constituem os alicerces de construção de todas as aplicações Qt, na medida em que todos os *widgets* Qt herdam `QObject`, assim como todas as suas características. Das inúmeras propriedades oferecidas pelo modelo, destacam-se as seguintes: mecanismo de comunicação directa entre objectos, designado *signals and slots*; sistema de eventos e filtro de eventos; a tradução de *strings* baseada no contexto como suporte à internacionalização; temporizadores; e árvores de objectos que permitem organizar hierarquicamente os objectos de uma maneira natural.

O mecanismo *signals and slots*, que permite a comunicação entre objectos `QObject`, é uma característica central da plataforma Qt e aquela que mais a diferencia em relação a outras plataformas. Normalmente na programação de interfaces gráficas, pretende-se que objectos de qualquer tipo, sobretudo *widgets*, possam comunicar uns com os outros. Noutras plataformas isto é conseguido recorrendo à passagem de apontadores para funções (*callbacks*), contudo o Qt possui uma técnica alternativa que é o uso de *signals and slots*. *Signals* (sinais) são emitidos por objectos sempre que determinado evento ocorre, ou seja, quando estes mudam de estado de um modo que pode ser considerado interessante para outros objectos. Um *slot*, por sua vez, é um método que é chamado automaticamente sempre que determinado sinal é detectado, podendo receber um número arbitrário de argumentos de qualquer tipo. A ligação entre os dois é feita através do método `connect`, que recebe como argumentos apontadores para o objecto emissor, o sinal emitido, o objecto receptor e o *slot* pretendido.

Outro mecanismo extremamente útil, disponibilizado pelo Qt, é o sistema de eventos. No Qt os eventos são objectos derivados da classe abstracta `QEvent` e representam acontecimentos ocorridos, tanto dentro da aplicação, como resultando de actividade externa sobre a qual a aplicação precisa de ser informada. Estes eventos podem ser recebidos, filtrados e processados por qualquer instância de uma subclasse `QObject`, destacando-se os eventos resultantes da actividade de periféricos, como o `QMouseEvent`, o `QKeyEvent` e ainda o `QTabletEvent`, que permite precisamente capturar e extrair informações da interacção de uma caneta com a superfície de um tablet.

Como referido anteriormente, os *widgets* são os principais elementos para a construção de interfaces gráficas em Qt. Os *widgets* permitem exibir informação, receber *input* do utilizador e actuam como *containers* para outros *widgets* que devam ser agrupados, ficando organizados hierarquicamente numa árvore de objectos. No Qt estes *widgets* são representados pela classe `QWidget`, que fornece a funcionalidade base de apresentar informação no ecrã e processar eventos de entrada gerados pelo utilizador. A classe `QWidget` estende `QObject`, herdando todas as propriedades da superclasse, algumas já mencionadas anteriormente, como a capacidade de

receber e filtrar eventos e o mecanismo *signals and slots*. A criação de *widgets* customizados assenta num sistema de heranças e subclasses de `QWidget`, que oferece uma enorme liberdade ao processo criativo, pois tanto permite estender directamente a classe genérica `QWidget` e produzir *widgets* de raiz, como permite estender e modificar qualquer um dos inúmeros *widgets* prefabricados da biblioteca Qt, garantindo assim que estes se adaptam aos diferentes cenários de cada aplicação.

Outro aspecto fundamental diz respeito ao posicionamento e organização dos diferentes *widgets* que compõem uma interface, assim como o aproveitamento do espaço disponível. Para além de permitir indicar explicitamente quais as coordenadas onde cada *widget* deve ser posicionado, método que apesar de garantir grande liberdade e precisão é demasiado inflexível e introduz demasiadas desvantagens ao desenvolvimento de interfaces, o Qt oferece ainda um sistema de *layouts* muito mais simples e poderoso para posicionar e organizar *widgets* no espaço. O sistema de *layouts* presente no Qt oferece uma forma simples, elegante e flexível de organizar automaticamente *widgets* dentro de um *widget* pai, garantindo que o espaço disponível é aproveitado da melhor maneira. Este sistema é definido por um conjunto de classes de gestão de *layouts* que são usadas para descrever como é que os *widgets* são dispostos na interface da aplicação. Estes *layouts* reposicionam e redimensionam automaticamente os *widgets* sempre que a quantidade de espaço disponível para eles é alterada, assegurando que eles são dispostos de forma consistente e que a interface como um todo permanece funcional. No Qt a gestão de *layouts* é assegurada sobretudo pelas classes, `QHBoxLayout`, `QVBoxLayout` e `QGridLayout`, que permitem dispor os *widgets* num linha horizontal, numa coluna vertical e numa grelha bidimensional, respectivamente. A construção de interfaces na plataforma Qt assenta sobretudo em combinações destas classes, que permitem criar estruturas complexas e dinâmicas de suporte aos *widgets* que compõem as diferentes interfaces criadas.

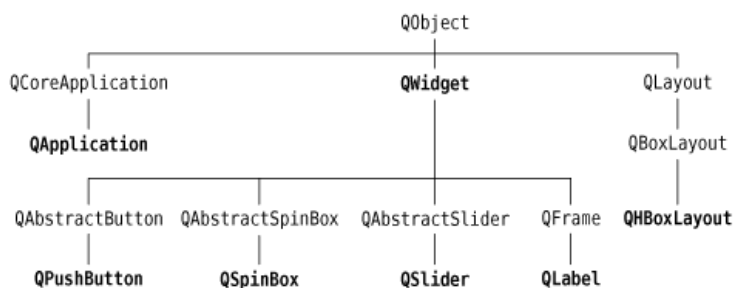


Figura 4.1: Visão geral da biblioteca Qt

A execução de uma aplicação Qt começa fora das classes, numa função especial denominada `main`. É a partir desta função, presente no ficheiro `main.cpp`, que toda a aplicação é construída. A função `main` da aplicação desenvolvida pode ser consultada na listagem 4.1.

Primeiro é criada uma instância da classe `QApplication`. Todas as aplicações desenvolvidas em Qt com interfaces gráficas têm de declarar obrigatoriamente um objecto do tipo `QApplication`. A classe `QApplication` é responsável pela gestão de recursos e controlo de fluxo de todas

as aplicações com interfaces gráficas e contém o ciclo principal de eventos, onde todos os eventos do sistema de janelas e de outras fontes são processados e encaminhados. A classe é ainda responsável pela inicialização e terminação da aplicação, pela gestão da sessão e pela maioria das configurações do sistema. De seguida é criado o *widget* principal da aplicação que actua como *container* para os restantes *widgets* que constituem a interface. Qualquer *widget* que não esteja incorporado dentro de outro *widget* pai, como acontece neste caso, é considerado como uma janela principal, sendo apontado pelo sistema como a raiz da árvore de objectos gráficos da interface dessa janela. Neste caso foi criada uma instância da classe `VirtualScriptorium`, o *widget* principal desta aplicação, que é uma subclasse de `QMainWindow`, que por sua vez é uma subclasse de `QWidget`. Depois de criada a janela principal da aplicação é chamado o método `showFullScreen()`, herdado de `QWidget`, que permite mostrar a janela principal da aplicação, ocupando por inteiro o ecrã. Finalmente, é chamado o método `exec()` do objecto `QApplication`, que faz com que a aplicação entre no seu ciclo de execução e espera até o método `exit()` ser chamado.

Listing 4.1: Função main da aplicação Scriptorium Virtual

```
1 int main(int argc, char *argv[]) {  
2     QApplication app(argc, argv);  
3     VirtualScriptorium vs;  
4     vs.showFullScreen();  
5     return app.exec();  
6 }
```

4.1.2 Arquitectura

A aplicação desenvolvida apresenta uma arquitectura organizada de acordo com as regras de funcionamento da plataforma Qt, no que diz respeito ao desenvolvimento de aplicações com interfaces gráficas. A arquitectura assenta numa estrutura em árvore, cuja raiz é a janela da aplicação e onde cada nó é um *widget* da interface, com todos os *widgets* presentes nessa estrutura a actuarem como *containers* para outros *widgets* e para toda a lógica associada ao seu funcionamento. A figura 4.2 apresenta a arquitectura da aplicação desenvolvida, onde se podem observar os principais *widgets* que a constituem, os elementos que suportam o seu funcionamento, assim como as relações estabelecidas entre eles.

Como referido anteriormente, a janela principal da aplicação é representada pela classe `VirtualScriptorium`, uma subclasse de `QMainWindow`, onde todos os componentes são criados e exibidos, e onde é feita a gestão da interface da aplicação. A classe `QMainWindow` oferece, como se pode observar na figura 4.3, uma estrutura para a construção de interfaces gráficas que inclui áreas de preenchimento facultativo para barras de menus, barras de ferramentas, barras de *status* e janelas secundárias, e uma área central de preenchimento obrigatório que pode albergar *widgets* de qualquer tipo.

Visto que a interface da aplicação não contempla quaisquer barras ou janelas auxiliares, somente uma área principal onde todos os componentes são expostos, é usada apenas a área

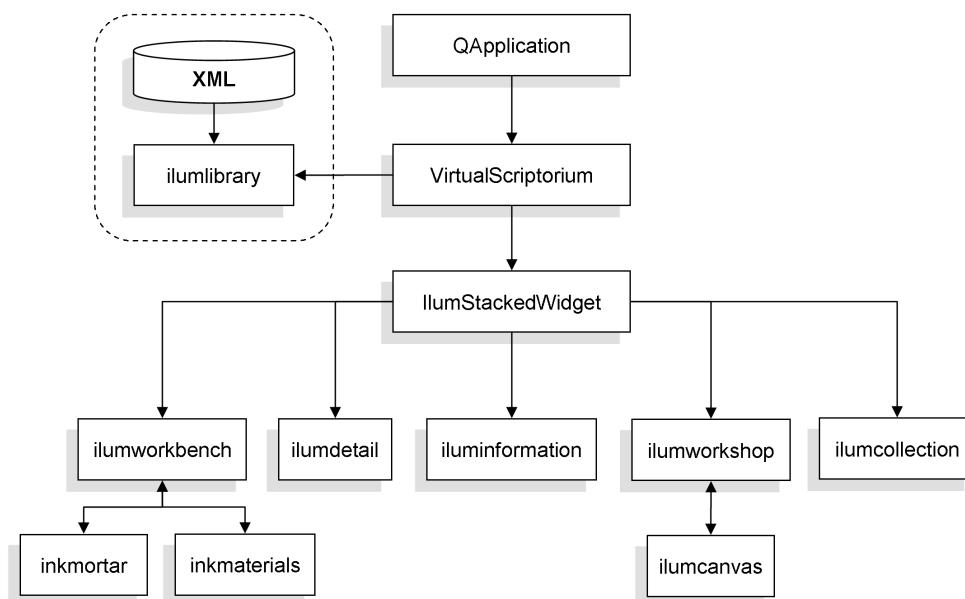


Figura 4.2: Arquitectura da aplicação Scriptorium Virtual

central da janela principal `VirtualScriptorium` para exibir as diferentes janelas que constituem a interface da aplicação. Estas janelas, que representam subconjuntos lógicos das funcionalidades da aplicação, para além de actuarem como *containers* para todos os *widgets* que constituem a sua interface, são também *widgets*, presentes na árvore de objectos, responsáveis pela lógica associada ao seu funcionamento, assim como pela gestão e coordenação de todos os *widgets* incluídos na sua estrutura. As janelas são representadas na aplicação por classes que estendem `QFrame`, uma subclasse de `QWidget`, que adiciona uma série de propriedades relacionadas com a aparência à classe que estende e que oferece as condições ideais para o pretendido: funcionalidades necessárias à customização da sua aparência e uma estrutura de suporte a todos os componentes incluídos em cada uma das janelas. Como também pode ser observado na figura 4.2, as janelas são representadas pelas seguintes classes:

ilumcollection: Janela onde são expostas as iluminuras disponíveis no sistema.

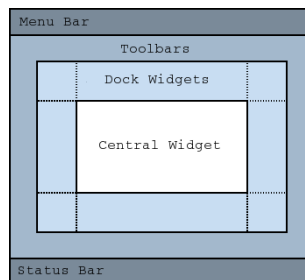
ilumworkshop: Janela onde são produzidas as iluminuras.

iluminformation: Janela onde pode ser consultada informação sobre a iluminura escolhida.

ilumdetail: Janela onde podem ser explorados os detalhes relativos à iluminura escolhida.

ilumworkbench: Janela onde são preparadas as tintas.

Associados a cada um destes *widgets*, encontram-se os diferentes processos de desenvolvimento que levaram à sua produção, com cada um a compreender a criação dos diferentes *widgets* que constituem cada uma das interfaces, assim como a lógica associada ao seu funcionamento. Estes processos são abordados individualmente mais à frente.

Figura 4.3: *Layout* da classe QMainWindow

O facto da interface da aplicação se encontrar compartimentada em diferentes janelas que preenchem completamente o ecrã, implica uma estrutura de suporte que permita gerir as janelas disponíveis e sempre que necessário seleccionar aquela que deve ser exibida na área central da janela principal da aplicação. A classe `QStackedWidget`, disponibilizada pela biblioteca Qt, oferece exactamente as funcionalidades pretendidas, uma pilha de *widgets* à qual podem ser adicionados quantos *widgets* forem necessários e um mecanismo que permite seleccionar exactamente qual é que deve ser apresentado. Contudo, apesar de bastante útil, a mudança de janelas oferecida pela classe `QStackedWidget` é demasiado abrupta e imediata para o que se pretende da interface da aplicação. Sempre que é despoletada uma transição entre janelas, pretende-se que essa mudança seja feita com alguma fluidez de forma a transmitir ao utilizador uma sensação de pertença e unidade entre todas as janelas que constituem a interface.

Para atingir este efeito criou-se a classe `IlumStackedWidget`, que estende a classe `QStackedWidget` e adiciona animação às transições entre janelas. Esta animação consiste no deslizamento da janela que se encontra actualmente no centro da janela principal da aplicação para fora da área visível e, simultaneamente, no mesmo movimento contínuo, o deslizamento da próxima janela para esse mesmo local. Isto permite criar a ilusão de ligação entre as duas janelas, pois durante a animação surgem as duas paralelamente na janela principal da aplicação, aparecendo a próxima janela por um lado e desaparecendo a antiga pelo lado oposto.

Isto é conseguido recorrendo à classe `QAbstractAnimation`, que é a base de todas as animações em Qt e oferece uma maneira fácil e eficaz de animar *widgets*. A classe `QAbstractAnimation` define as propriedades e funcionalidades comuns a todas as animações, nomeadamente, a capacidade de iniciar, parar e pausar uma animação. Uma das classes que herda estas características é a classe `QPropertyAnimation`, a classe responsável pela animação de *widgets* através da interpolação entre os valores definidos para as propriedades iniciais e finais da animação. A criação das animações para as duas janelas pode ser consultada na listagem 4.2.

Listing 4.2: Criar animações em Qt

```

1  /* Criar animação para a janela actual */
2  QPropertyAnimation *animnow = new QPropertyAnimation(widget(now), "pos");
3  animnow->setDuration(m_speed);
4  animnow->setEasingCurve(m_automationtype);
5  animnow->setStartValue(QPoint(pnow.x(), pnow.y()));
6  animnow->setEndValue(QPoint(offsetx+pnow.x(), offsety+pnow.y()));

```

```

7
8  /* Criar animação para a próxima janela */
9  QPropertyAnimation *animnext = new QPropertyAnimation(widget(next), "pos");
10 animnext->setDuration(m_speed);
11 animnext->setEasingCurve(m_aniimationtype);
12 animnext->setStartValue(QPoint(-offsetx+pnnext.x(), offsety+pnnext.y()));
13 animnext->setEndValue(QPoint(pnnext.x(), pnnext.y()));
14
15 /* Criar grupo de animações paralelas a partir das animações anteriores */
16 QParallelAnimationGroup *animgroup = new QParallelAnimationGroup;
17 animgroup->addAnimation(animnow);
18 danimgroup->addAnimation(animnext);
19
20 /* Iniciar animação */
21 QObject::connect(animgroup, SIGNAL(finished()),this,SLOT(animationDoneSlot()));
22 animgroup->start();

```

Primeiro é criada a animação para a janela que se encontra em exibição e em segundo lugar é criada a animação para a janela que vai substituir a primeira. As animações são criadas fundamentalmente da mesma maneira, variando apenas os *widgets* afectados e as propriedades iniciais e finais definidas para as interpolações. Em primeiro lugar é criada uma instância da classe `QPropertyAnimation`, passando no construtor o *widget* que será animado e o nome da propriedade. De seguida são chamados os métodos `setDuration` e `setEasingCurve` para definir o tempo de duração em milissegundos e a curva de atenuação usada na animação, respectivamente.

A atenuação permite acelerar ou retardar o início ou fim de uma animação para movimentos mais realistas ou efeitos mais agradáveis. A atenuação é uma curva matemática que é aplicada aos valores de propriedade de uma interpolação. O efeito final da interpolação é o resultado da combinação do intervalo dos valores de propriedade da interpolação e da curva de atenuação. No Qt estas curvas são representadas pela classe `QEasingCurve`, que permite escolher exactamente a curva de atenuação usada, de um conjunto alargado de tipos de curvas.

Finalmente, são definidas as propriedades iniciais e finais para a interpolação através dos métodos `setStartValue` e `setEndValue`, respectivamente. Neste caso as propriedades foram definidas como instâncias da classe `QPoint` e representam as posições iniciais e finais das janelas. Estas posições são definidas previamente de acordo com a direcção escolhida para a animação, podendo as janelas deslizar de cima para baixo, de baixo para cima, da direita para a esquerda e da esquerda para a direita.

Sempre que é necessário animar vários *widgets* relacionados em simultâneo, recorre-se às subclasses de `QAnimationGroup`, nomeadamente, `QSequentialAnimationGroup` e `QParallelAnimationGroup`. Estas subclasses actuam como *containers* para outras animações de maneira a que estas possam ser animadas em sequência ou paralelamente, respectivamente. Neste caso pretende-se que as duas janelas sejam animadas paralelamente. A criação do grupo de animações para as duas janelas pode ser consultada na listagem 4.2.

Em primeiro lugar é criada uma instância da classe `QParallelAnimationGroup` e são adicionadas as animações previamente criadas ao objecto. Em segundo lugar é usado o mecanismo *signals and slots* para ligar o sinal emitido pela conclusão da animação, ao *slot* `animationDoneSlot`. Isto garante que quando a animação termina é chamado o método `animationDoneSlot` que, por sua vez, chama o método `setCurrentIndex`, herdado de `QStackedWidget`. Este método é responsável pela efectivação da mudança real das janelas na pilha de `QStackedWidget`, garantindo que a janela escolhida passa a ser de facto a janela visível no centro da janela principal da aplicação. Finalmente, é chamado o método `QAbstractAnimation::start()` que inicia a animação.

Apesar de tudo, a classe `IllumStackedWidget` continua a ser essencialmente uma subclasse de `QWidget`, herdando todas as suas características e comportamentos, aliás, é a classe `IllumStackedWidget` que é adicionada directamente à área central da janela principal da aplicação, actuando como *container* para todas as janelas incluídas na interface. A janela principal `VirtualScriptorium` cria todas as janelas, assim como uma instância de `IllumStackedWidget`, depois adiciona todas as janelas criadas à pilha de `IllumStackedWidget` e adiciona `IllumStackedWidget` à sua área central. A classe `IllumStackedWidget` actua como um intermediário entre a janela principal da aplicação e as janelas que constituem a interface, sendo responsável pela gestão da pilha de *widgets*, pela exibição do *widget* seleccionado e pela animação das transições.

As transições entre janelas ficam a cargo da classe `VirtualScriptorium`, recorrendo ao mecanismo *signals and slots*. Sempre que numa das janelas é desencadeada uma transição, é emitido um sinal que se encontra ligado a um *slot* de `VirtualScriptorium`. Estes *slots* executam depois as operações necessárias à transição, acedendo aos métodos disponibilizados pela classe `IllumStackedWidget`. Essencialmente, é enviado um sinal até à raiz da árvore de objectos, que depois procede às mudanças no seu descendente directo, a classe `IllumStackedWidget`.

Para além de `IllumStackedWidget`, pode também ser observado na figura 4.2 um componente responsável pelo carregamento de toda a informação armazenada em ficheiros XML, necessária ao funcionamento da aplicação. O desenvolvimento deste componente pode ser consultado em maior detalhe na secção 4.1.3. A aplicação inclui ainda um mecanismo de *logging* que regista todas as interacções dos utilizadores com a aplicação e que serve de suporte à avaliação do comportamento da aplicação em situações de utilização real.

A inicialização de todos estes componentes é efectuada no construtor da classe `VirtualScriptorium` de acordo com o código apresentado na listagem 4.3.

Listing 4.3: Inicialização e configuração da aplicação Scriptorium Virtual

```

1  /* Carregar dados a partir dos ficheiros XML */
2  library = new ilumlibrary(QCoreApplication::applicationDirPath() + "/data.xml");
3
4  /* Criar janelas da aplicação */
5  ilumCollection = new ilumcollection(library);
6  ilumWorkshop = new ilumworkshop;
7  ilumInformation = new iluminformation;
8  ilumDetailsPanel = new ilumdetail;
9  ilumWorkbench = new ilumworkbench;
```

```
10
11 /* Criar instância de IlumStackedWidget */
12 ilumStackedWidget = new IlumStackedWidget();
13
14 /* Adicionar todas as janelas criadas a ilumStackedWidget */
15 ilumStackedWidget->addWidget(ilumCollection);
16 ilumStackedWidget->addWidget(ilumWorkshop);
17 ilumStackedWidget->addWidget(ilumInformation);
18 ilumStackedWidget->addWidget(ilumDetailsPanel);
19 ilumStackedWidget->addWidget(ilumWorkbench);
20
21 /* Definir ilumStackedWidget como o widget central de VirtualScriptorium */
22 setCentralWidget(ilumStackedWidget);
```

4.1.3 Colecção de iluminuras

Como referido anteriormente em 3.1.1, todas as iluminuras presentes na aplicação são representadas no sistema por imagens. Estas imagens contêm grande parte da informação disponível sobre as iluminuras presentes na aplicação, desde informações gerais sobre as iluminuras, até aos processos de produção e preparação das tintas. Toda a informação textual disponibilizada na aplicação é apresentada também sob a forma de imagens, o que permite respeitar a estética da aplicação, garantindo uma apresentação mais agradável dos textos. A produção destes conteúdos decorreu paralelamente à produção da instalação interactiva e ficou a cargo dos responsáveis do Departamento de Conservação e Restauro.

Apesar destas imagens representarem grande parte da informação disponível na aplicação, sozinhas são insuficientes no que concerne à especificação dos processos envolvidos na produção das iluminuras e preparação das tintas. As imagens permitem exibir uma série de informações sobre as iluminuras, como fotografias, figuras auxiliares, inclusive informação textual, contudo, sozinhas são insuficientes para especificar exactamente os passos necessários à produção das iluminuras. Assim, a especificação destes processos e a organização de todas as imagens disponíveis no sistema é efectuada, de forma estruturada, num ficheiro XML. Este ficheiro apresenta uma estrutura que pode ser observada em maior detalhe no anexo E, através da consulta do respectivo XML Schema.

Esta solução, para além de permitir organizar e complementar a informação oferecida pelas imagens, evita que o seu caminho seja embutido directamente no código da aplicação (*hard-coded*), contribuindo para a separação entre o desenvolvimento da aplicação e a produção dos conteúdos e simplificando substancialmente modificações e a integração de novas iluminuras no sistema. A aplicação foi desenvolvida de forma a garantir que a integração de novas iluminuras no sistema seja relativamente simples, evitando quaisquer alterações ao código da aplicação. A adição implica somente o armazenamento das novas imagens no respectivo repositório e a especificação das novas iluminuras no ficheiro XML. Desde que as novas iluminuras respeitem as regras de validação impostas pelo XML Schema a aplicação não terá qualquer problema em carregá-las para o sistema. Neste aspecto a aplicação é extremamente genérica, carregando

um número arbitrário de iluminuras do ficheiro XML e garantindo que todas oferecem as mesmas funcionalidades aos utilizadores. Este método oferece ainda suporte à internacionalização da aplicação, na medida em que é possível criar um ficheiro XML para cada uma das línguas pretendidas e carregar apenas aquele que corresponde à língua desejada.

O ficheiro XML é carregado por um componente específico da aplicação, que faz a análise e interpretação do ficheiro XML e mapeia os dados extraídos para um conjunto de classes organizadas de forma idêntica à do ficheiro XML, como pode ser observado no diagrama de classes da figura 4.4. Esta informação é carregada para estes objectos, que são depois disponibilizados à camada de visualização, onde são consumidos e a informação é apresentada aos utilizadores.

Para além da informação que é consumida pela aplicação, é importante também mencionar a informação que é produzida, nomeadamente as iluminuras que são criadas pelos utilizadores. Após a sua conclusão, as iluminuras são armazenadas num repositório, que depois é carregado directamente para a aplicação multitoque, ficando disponíveis para consulta a todos os visitantes numa galeria produzida para o efeito.

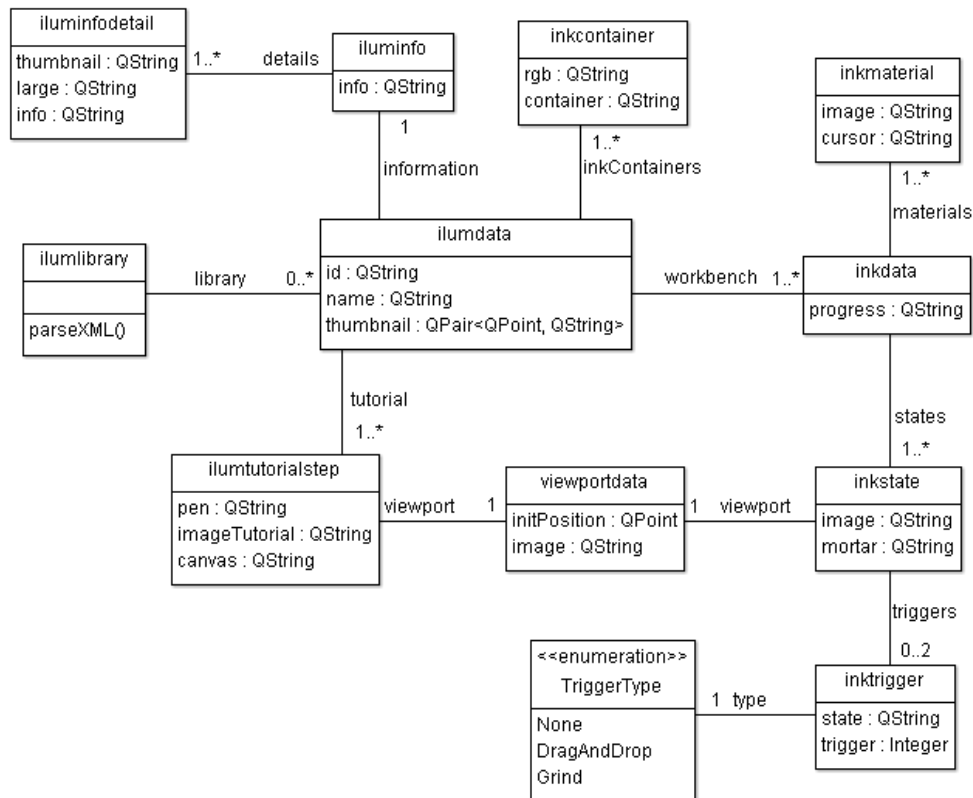


Figura 4.4: Diagrama de classes da aplicação Scriptorium Virtual

4.1.4 Selecção de iluminuras

A janela de selecção de iluminuras é representada pela classe *ilumcollection*, que estende *QFrame*, e, como foi referido em 3.1.2.1, é composta por uma série de *thumbnails* que representam todas as iluminuras presentes no sistema. Estes *thumbnails* encontram-se dispersos por toda a área

disponível na interface e são carregados directamente do ficheiro XML, como apresentado na listagem 4.4.

Listing 4.4: *Thumbnail* de iluminura especificado no ficheiro XML

```
1 <data>
2   <ilum id="ilum01">
3     <name>A Pomba e o Falcão</name>
4     <thumbnail x="468" y="724">/resources/ilums/ilum01/thumbnail.png</thumbnail>
5     ...
6   </ilum>
7   ...
8 </data>
```

Deste ficheiro são extraídas as posições iniciais dos *thumbnails*, assim como os caminhos das imagens circulares usadas para retratar as iluminuras disponíveis. Esta informação é armazenada nas respectivas instâncias da classe *ilumdata* (figura 4.4), que depois são disponibilizadas à camada de visualização para que esta carregue todas as imagens para objectos do tipo *QLabel* e as posicione nos locais previamente definidos. A classe *QLabel* é usada para exibir diferentes tipos de informação (e.g., texto, imagens, vídeo) e possui todas as propriedades e características herdadas de *QFrame*.

Apesar das posições iniciais predefinidas, é possível arrastar os *thumbnails* para qualquer local da interface. Esta funcionalidade foi desenvolvida recorrendo ao mecanismo de eventos oferecido por todas as classes que estendam *QWidget*. Como já foi mencionado, todas as janelas presentes na aplicação estendem *QFrame*, uma subclasse de *QWidget*, logo é possível reimplementar em qualquer janela os métodos associados à detecção e tratamento de eventos. Neste caso em particular foram reimplementados os métodos *mousePressEvent* e *dropEvent*, essenciais ao desenvolvimento do efeito arrastar e largar (*drag and drop*) na plataforma Qt. O método *mousePressEvent* é chamado sempre que são detectados eventos despoletados pelo pressionar de um botão do rato, neste caso respondendo também ao toque da caneta na superfície do tablet. O método *dropEvent*, por sua vez, é chamado sempre que um objecto do tipo *QDrag* é largado dentro de outro *widget*. A abordagem suportada pela plataforma Qt para o mecanismo *drag and drop* assenta fundamentalmente no conceito de copiar e colar usado recorrentemente na nossa interacção diária com computadores para transferir informação. São capturadas todas as informações relativas ao *widget* que vai ser arrastado para uma estrutura de dados, o *widget* original é escondido, é criada uma representação temporária do *widget* para o arrastamento e no fim é criado um novo *widget* com a informação capturada e o original é apagado.

A interacção começa quando o utilizador toca com a ponta da caneta na superfície do tablet. Nesse instante é chamado o método *mousePressEvent*, que começa por verificar se o utilizador tocou ou não sobre um dos *thumbnails*. Em primeiro lugar é verificado se existe algum *widget* do tipo *QLabel* na coordenada correspondente ao toque da caneta e finalmente é verificado se essa coordenada se encontra dentro da área circular do *thumbnail*. Caso o utilizador tenha de facto tocado em cima de um *thumbnail*, é armazenada essa coordenada para verificação

posterior e é carregada a imagem do *thumbnail* para uma instância da classe *QPixmap*, uma classe otimizada para o carregamento e exibição de imagens. Esta imagem é carregada para uma instância da classe *QMimeData*, uma classe que permite o armazenamento de informação e a sua transferência entre diferentes aplicações, janelas ou *widgets*, ideal para a funcionalidade *drag and drop* desenvolvida e que assenta fortemente no conceito cortar e colar mencionado anteriormente. Para além da imagem, é carregada também a distância entre o local onde o toque da caneta ocorreu e a posição inicial do *thumbnail*, de forma a posicionar correctamente o *thumbnail* na sua futura posição.

De seguida, é criada uma instância da classe *QDrag*, onde é armazenado o objecto do tipo *QMimeData* acabado de criar, é usado o *QPixmap* criado anteriormente para representar o *thumbnail* durante o arrastamento e é escondida a *QLabel* original recorrendo a um novo *QPixmap* transparente.

Finalmente, é chamado o método *exec()* da classe *QDrag*, que inicia efectivamente a operação de arrastamento. Como argumento, é definido um conjunto de acções para o objecto *QDrag*, que informa todos os possíveis receptores do novo *widget* quais são as acções suportadas para esta transacção. Este mecanismo permite dar *feedback* ao método que inicia o processo sobre o local onde o *widget* foi libertado e a forma como a operação foi concluída, para que este possa agir de acordo com o definido. Isto é útil pois o programa não tem maneira de saber o que o utilizador vai fazer antes de a operação estar efectivamente concluída. O método *exec()* permanece bloqueado até à recepção do *feedback*, que só é enviado quando o utilizador liberta o *thumbnail* na interface. As acções definidas foram: *Qt::MoveAction* e *Qt::CopyAction*. Caso *exec()* retorne *MoveAction*, o arrastamento é dado como concluído e o *widget* original é destruído, caso retorne *CopyAction*, o arrastamento é dado como incompleto e é restituído o *widget* original, voltando tudo ao ponto inicial.

Assim que o utilizador liberta o *thumbnail*, é verificado pelo Qt sobre qual *widget* é que este foi libertado. Para que qualquer *widget* possa aceitar eventos deste tipo é necessário chamar no seu construtor o método *setAcceptDrops(true)*. Neste caso, tendo em conta que a janela preenche totalmente o ecrã, o único local onde esta pode ser libertada é precisamente dentro de *ilumcollection*, o que desencadeia a chamada do método reimplementado *dropEvent*, responsável pela conclusão da operação.

O método *dropEvent* começa por descarregar toda a informação armazenada no evento, carregada previamente aquando da inicialização da operação, nomeadamente, a imagem do *thumbnail* e a distância entre o local onde o toque da caneta ocorreu e a posição inicial do *thumbnail*. De seguida, é criada uma nova *QLabel* a partir da imagem descarregada, criando assim uma cópia do *thumbnail* original, que é posicionada no seu novo local com o auxílio da distância descarregada. Esta distância permite compensar a posição da caneta dentro do *thumbnail* e posicioná-lo correctamente na sua nova posição. Segue-se a definição da acção que será retornada em *QDrag::exec()* no método *mousePressEvent*, que por sua vez desencadeará o procedimento adequado para a conclusão da operação. Neste caso é retornado *MoveAction*, indicativo do sucesso da operação.

Tendo em conta que a função desta janela é fundamentalmente a selecção da iluminura que se pretende reproduzir, foi necessário criar um mecanismo que permitisse distinguir entre uma operação de arrastamento e uma operação de selecção. Esta distinção é feita no método `dropEvent`, depois de concluída a operação de arrastamento, e consiste essencialmente na comparação entre a posição inicial e final de contacto da caneta com a superfície do tablet. A posição inicial é armazenada previamente no método `mousePressEvent` e a posição final é obtida directamente do evento recebido pelo método `dropEvent`. Caso as posições sejam diferentes, significa que o *thumbnail* foi arrastado, se as posições forem iguais significa que foi seleccionada a iluminura correspondente ao *thumbnail*. Teoricamente esta mecânica deveria funcionar, contudo na prática verificou-se que não resulta de acordo com o esperado. Constatou-se que é preciso distinguir também entre os casos em que o utilizador se limita a arrastar os *thumbnails* pela interface e os casos em que o utilizador tenta seleccionar uma iluminura, mas devido às limitações inerentes ao dispositivo de interacção usado, acaba por arrastar acidentalmente o *thumbnail*. O reduzido atrito apresentado pela superfície destes dispositivos conduz a que seja extremamente complicado tocar com a caneta na sua superfície sem qualquer tipo de arrastamento. Desta forma é necessário testar se a diferença entre as posições se encontra dentro de uma margem de erro (*threshold*) considerada aceitável para este problema. Procede-se então ao cálculo da diferença entre a posição inicial e final, caso esta seja superior ao *threshold*, considera-se uma operação normal de arrastamento e o método `dropEvent` termina sem mais operações, caso seja inferior ao *threshold*, considera-se uma operação de selecção e é emitido um sinal ligado a um *slot* da janela principal `VirtualScriptorium`, que se encarrega do carregamento da iluminura escolhida e da transição para a janela de produção de iluminuras.

4.1.5 Produção de iluminuras

Esta janela é representada pela classe `ilumworkshop`, que estende `QFrame` e é responsável por todos os componentes que compõem a sua interface, pela sua organização e pelos mecanismos associados ao seu funcionamento. Todos os componentes que compõem a interface são criados e organizados quando a aplicação é inicializada, contudo permanecem sem conteúdo até ao momento em que é seleccionada uma iluminura para produção. Quando é seleccionada uma iluminura, é emitido um sinal na classe `ilumcollection` com a identificação da iluminura escolhida. Este sinal encontra-se ligado a um *slot* da janela principal `VirtualScriptorium`, responsável por extrair a informação relativa à iluminura escolhida com o identificador recebido, desencadear o carregamento do conteúdo para os respectivos *widgets* e pela transição para a janela de produção de iluminuras. O carregamento é feito através da chamada do método `loadWorkshop` da classe `ilumworkshop`, que recebe como argumento um objecto do tipo `ilumdata`, com todas as informações associadas à iluminura escolhida, o que inclui os passos do tutorial e definição das tintas. O método `loadWorkshop` armazena localmente esta informação e procede à criação dos botões das tintas e ao carregamento do primeiro passo do tutorial.

Os botões, representados na interface por instâncias da classe `QPushButton`, são criados a partir das imagens dos recipientes e dos códigos RGB correspondentes a cada uma das tintas.

O primeiro passo do tutorial, assim como todos os restantes, contém informações relativas às definições da caneta e ao caminho das imagens que surgem na interface para auxiliar o utilizador, nomeadamente, uma imagem com informação textual, uma imagem com informação complementar e uma imagem auxiliar que surge na área de desenho e pintura.

A listagem 4.5 apresenta um exemplo de um passo do tutorial.

Listing 4.5: Passo da produção de iluminuras especificado no ficheiro XML

```
1 <tutorial>
2   <step>
3     <!-- Imagem com informação complementar -->
4     <viewport x="100" y="150">viewport.png</viewport>
5     <!-- Imagem com informação textual -->
6     <imagetutorial>imagetutorial.png</imagetutorial>
7     <!-- Definição da caneta -->
8     <pen>BrushFill</pen>
9     <!-- Imagem auxiliar de desenho -->
10    <canvas>canvas.png</canvas>
11  </step>
12  ...
13 </tutorial>
```

O carregamento de passos do tutorial consiste essencialmente em modificar as propriedades da caneta de acordo com o definido e carregar as imagens para os respectivos *widgets*. Este processo é igual, tanto para o carregamento do primeiro passo, como para o carregamento dos restantes, que é executado sempre que se dá uma mudança de passo. As mudanças de passos são desencadeadas explicitamente pelo utilizador, sempre que toca num dos botões definidos para tal. O controlo de todos os botões presentes na interface é realizado com recurso ao mecanismo *signals and slots*, que permite ligar os sinais emitidos por cada um dos botões a *slots* que executam as operações necessárias, neste caso métodos que desencadeiam o carregamento do anterior/próximo passo do tutorial.

Como foi referido anteriormente em 3.1.2.2, a interface encontra-se organizada em três áreas principais: o tutorial da iluminura escolhida, os recipientes das tintas e a área de desenho e pintura.

A área do tutorial é composta por uma instância da classe `QLabel`, que exibe as imagens com informações textuais e por um componente que permite apresentar num espaço bastante reduzido imagens de elevada resolução com os detalhes da iluminura escolhida. Este último componente permite ao utilizador navegar e explorar os detalhes presentes na imagem, arrastando-a com a caneta e com o auxílio de *kinetic scrolling*. A base deste componente é a classe `QScrollArea`, que permite a exibição de outro *widget* dentro da sua área visível e, caso este exceda as dimensões dessa área, a adição de barras de *scrolling* para que toda a área do *widget* filho possa ser observada. Neste caso o *widget* adicionado à área visível de `QScrollArea` é um objecto `QLabel`, de modo a carregar e exibir as imagens de elevada resolução disponíveis.

Contudo, apesar de esta classe oferecer as funcionalidades base para o pretendido, não permite a navegação no *widget* filho através do arrastamento da caneta, apenas através das barras

de *scrolling*. Para além disso, não inclui *kinetic scrolling*, que proporciona uma interacção mais agradável aos utilizadores. Desenvolveu-se então uma classe auxiliar Viewport, que permite modificar e adicionar à classe QScrollArea as funcionalidades pretendidas para a aplicação. A classe Viewport é responsável por todas as propriedades e métodos relacionados com o *kinetic scrolling*, assim como todas as alterações e operações desencadeadas na classe base QScrollArea. Depois de criada uma instância da classe Viewport, é chamado o método activateOn, que recebe como argumento um apontador para um objecto que estenda QAbstractScrollArea, neste caso o objecto QScrollArea usado, ao qual se deseja aplicar as funcionalidades da classe Viewport. O método activateOn esconde as barras de *scrolling* do objecto QScrollArea recebido e instala um filtro de eventos nesse mesmo objecto.

Um filtro de eventos é fundamentalmente um objecto que recebe todos os eventos que têm como destinatário outro objecto, actuando como intermediário entre a entidade que despoleta os eventos e os seus receptores. O filtro de eventos processa os eventos capturados e decide se os deve filtrar, ou seja, parar, ou reencaminhar para o seu destinatário original. Para criar um filtro de eventos é necessário criar uma classe que estenda QObject e reimplementar o método eventFilter. Neste caso foi criada a classe Viewport, que actua como o filtro de eventos que é instalado no objecto QScrollArea recebido. Deste modo, todos os eventos enviados para o objecto QScrollArea são reencaminhados para a classe Viewport. O filtro de eventos da classe Viewport é responsável por filtrar todos os eventos despoletados pela interacção da caneta com a área visível de QScrollArea e por reencaminhar os restantes para o seu destinatário original. São filtrados os seguintes eventos: MouseButtonPress, MouseButtonRelease e MouseMove. Com base nestes eventos é possível à classe Viewport capturar todas as interacções do utilizador com área de QScrollArea e adicionar-lhe o efeito pretendido.

Enquanto os eventos filtrados forem do tipo MouseButtonPress, seguidos de vários eventos do tipo MouseMove, significa que o utilizador se encontra nesse instante a arrastar com a caneta o conteúdo de QScrollArea. Nestas situações o filtro limita-se a armazenar a posição actual da caneta para cálculo futuro da velocidade e a deslocar o conteúdo para as posições indicadas directamente pelo utilizador. Este reposicionamento é feito através da redefinição dos valores associados às barras de *scrolling* de QScrollArea que, apesar de escondidas, permanecem activas e são a único meio para reposicionar o seu conteúdo. Quando surge um evento do tipo MouseButtonRelease significa que o utilizador levantou a caneta da superfície do tablet e parou o arrastamento manual do conteúdo de QScrollArea, começando nesse instante o arrastamento automático associado ao *kinetic scrolling*.

Primeiro é calculada a velocidade de arrastamento no momento exacto em que o utilizador levanta a caneta da superfície do tablet. Este cálculo consiste na diferença entre a posição armazenada previamente, que corresponde à posição extraída do último evento MouseMove recebido, e a última posição conhecida na superfície do tablet, ou seja, a posição extraída do último evento MouseButtonRelease recebido. Depois de calculada a velocidade inicial, é-lhe aplicada iterativamente uma desaceleração constante, de modo a reduzir progressivamente o seu valor e a criar o efeito pretendido. É com base nesta velocidade inicial e na constante de desaceleração que é interpolado o arrastamento automático do conteúdo de QScrollArea. O

cálculo da desaceleração é apresentado na listagem 4.6.

Listing 4.6: Método para calcular a desaceleração associada ao efeito *kinetic scrolling*

```
1 int x = qBound(-max, speed.x(), max); //Delimitar velocidade x
2 int y = qBound(-max, speed.y(), max); //Delimitar velocidade y
3 x = (x == 0) ? x : (x > 0) ? qMax(0, x - a) : qMin(0, x + a); //Desaceleração x
4 y = (y == 0) ? y : (y > 0) ? qMax(0, y - a) : qMin(0, y + a); //Desaceleração y
5 return QPoint(x, y);
```

O cálculo é feito em duas fases. Primeiro é delimitada a velocidade calculada anteriormente de maneira a descartar valores demasiado elevados. O método `qBound` retorna a velocidade delimitada entre o valor mínimo e máximo, definidos inicialmente como constantes e recebidos como argumento com a velocidade calculada anteriormente. Finalmente, é calculada a velocidade final a partir dos valores anteriormente delimitados e da constante definida inicialmente para a desaceleração. Caso a velocidade seja nula, significa que o movimento terminou e não é feita qualquer alteração à velocidade. Caso a velocidade seja positiva, é subtraída à velocidade a constante de desaceleração. Por fim, caso a velocidade seja negativa, é adicionada à velocidade a constante de desaceleração. Este cálculo é efectuado iterativamente até que o valor da velocidade atinja valores nulos, momento em que o arrastamento automático termina. É ainda importante mencionar que o utilizador pode interagir com a área de `QScrollArea` a qualquer momento, mesmo durante um arrastamento automático, resultando sempre no início de uma nova interacção.

Como foi referido anteriormente, para além da informação indispensável ao carregamento dos *widgets* presentes nesta área, os passos do tutorial compreendem também uma imagem que permite auxiliar o utilizador durante as diferentes fases que compõem a produção da iluminura escolhida. Esta imagem surge, não na área do tutorial, mas sim na área de desenho e pintura, onde o utilizador pode efectivamente reproduzir a iluminura escolhida, recorrendo à caneta e seguindo todas as indicações oferecidas pelo tutorial. A área de desenho e pintura é composta por dois componentes sobrepostos. Uma instância da classe `QLabel`, responsável pela exibição das imagens auxiliares, e um componente onde o utilizador pode desenhar e pintar, responsável pela captura de todas as interacções da caneta dentro desta área, pelo seu processamento e pela tradução dessas interacções em pontos, linhas e formas. Estes componentes encontram-se sobrepostos para recriar a utilização de papel vegetal no *workshop*. Quando a instância de `QLabel` tem uma imagem, o fundo do componente onde o utilizador desenha e pinta é transparente, assim o utilizador pode desenhar directamente por cima da imagem e copiar todos os pormenores para a sua iluminura. Quando não existe imagem auxiliar, é restituído o fundo original com uma cor semelhante à do pergaminho.

O *widget* onde o utilizador desenha e pinta é representado na aplicação pela classe `ilum-canvas`. Esta classe, que estende `QWidget`, representa conceptualmente o pergaminho onde as iluminuras eram originalmente produzidas e é responsável por todas propriedades associadas à produção da iluminura e pela captura e tradução de todas as interacções realizadas na área de desenho com a caneta pelo utilizador. Algumas destas propriedades encontram-se associadas à tinta usada para desenhar e pintar. A tonalidade da cor que é aplicada na iluminura

encontra-se dependente de três factores: a tinta escolhida pelo utilizador, a quantidade de tinta disponível e a pressão exercida pela caneta na superfície do tablet.

A tinta é escolhida manualmente pelo utilizador através do toque num dos botões presentes junto à área de desenho e pintura. Sempre que isto acontece é despoletado um sinal ligado a um *slot* de *ilumworkshop*, que recebe como argumento o código RGB da tinta escolhida e procede à alteração da respectiva propriedade em *ilumcanvas*, representada nessa classe por uma instância de *QColor*.

Um dos objectivos desta aplicação é transmitir aos utilizadores parte da experiência do *workshop*. Uma das particularidades dessa experiência, inerente à actividade de pintar e desenhar, é a necessidade de refrescar periodicamente a tinta usada nos diferentes instrumentos de pintura, graças ao consumo resultante da sua utilização. De modo a capturar esta particularidade, a classe *ilumcanvas* possui uma propriedade associada à quantidade de tinta disponível, que visa precisamente a recriação dos principais aspectos envolvidos na utilização de tinta numa situação real. Sempre que o utilizador toca num dos botões de selecção de tinta, para além de alterar a cor da tinta usada, reinicializa também a quantidade de tinta disponível. Posteriormente, todas as interacções do utilizador com a área de pintura resultam no consumo da tinta disponível. Este consumo encontra-se directamente relacionado com a distância percorrida e a pressão exercida pela ponta da caneta na superfície do tablet.

A quantidade de tinta disponível tem uma influência directa na tonalidade da cor que é aplicada na iluminura. Visualmente esta propriedade influencia a tonalidade da cor, pois é co-responsável pela definição do nível de opacidade que é aplicado à cor da tinta escolhida, antes da sua utilização. O controlo do nível de opacidade permite simular o efeito produzido pela utilização da tinta numa situação real, onde o seu consumo contínuo resulta numa perda incremental de intensidade e na mistura dos seus tons com o fundo onde está a ser aplicada. A quantidade de tinta é representada em *ilumcanvas* por uma variável com valores compreendidos entre 0 e 255, com o valor mínimo a significar transparência total e o valor máximo a representar uma cor totalmente opaca. Este intervalo de valores permite criar uma correspondência directa entre a quantidade de tinta disponível e o nível de opacidade que é definido no canal alfa da classe *QColor*. O código RGB obtido aquando da selecção da tinta, mais o nível de opacidade definido, formam o código RGBA da cor que depois é aplicada à iluminura.

Para além da quantidade de tinta, outro factor que influencia a opacidade é a pressão exercida pela caneta na superfície do tablet. Quanto maior é a pressão, mais elevada é a opacidade da cor usada na produção da iluminura. Esta funcionalidade permite ao utilizador controlar de forma precisa, natural e sem botões, o nível de opacidade das cores empregadas na área de desenho, capturando parte das subtilezas inerentes à pintura com instrumentos reais, onde a pressão exercida na área de desenho influencia fortemente a intensidade da cor utilizada. Para além disso, viabiliza a mistura das cores empregadas com as cores já existentes na iluminura, técnica que enriquece o processo de produção graças à expressividade criativa que oferece.

Para além da opacidade, a pressão influencia também o consumo da tinta e a espessura do traço da caneta. Quanto maior é a pressão exercida pela caneta, mais tinta é consumida e elevada é a espessura dos traços desenhados. A espessura dos traços é influenciada também

pelo tipo de caneta que está a ser utilizado no momento, que pode ser diferente para cada passo do tutorial. Em conjunto, estas propriedades são responsáveis pela configuração de uma instância da classe QPen, presente em *ilumcanvas* para representar as definições associadas à caneta utilizada pelo utilizador. A classe QPen representa conceptualmente um instrumento de pintura, englobando todas as propriedades associadas a um instrumento desta natureza, onde se destacam a cor, a espessura, o estilo do traço e o modo como as uniões entre traços são desenhadas. Com base nestas propriedades, a classe QPen define como é que um objecto do tipo QPainter deve desenhar as linhas, os contornos e as formas criadas pelo utilizador.

A classe QPainter permite desenhar e pintar em *widgets* e em outros dispositivos de pintura, oferecendo métodos optimizados para realizar a maioria da pintura necessária a programas que requeiram essa funcionalidade, permitindo desenhar desde linhas simples, até formas mais complexas, como texto e imagens. O uso mais comum da classe QPainter é dentro de um evento de pintura, onde é construído, configurado com um objecto QPen e depois usado para pintar o pretendido. Os eventos de pintura, ou QPaintEvent, são emitidos pela plataforma sempre que é necessário actualizar o aspecto de determinado *widget*. O método paintEvent, responsável pelo tratamento deste tipo de evento, pode ser reimplementado em qualquer subclasse de QWidget ou QObject, tal como todos os eventos oferecidos pela plataforma Qt.

Nesta aplicação, a classe QPainter é usada para desenhar todos os traços criados pelo utilizador com a caneta para duas instâncias da classe QPixmap, uma inicializada com fundo transparente e outra com fundo de pergaminho. Deste modo é possível comutar rapidamente e de forma eficiente entre os dois fundos disponíveis, consoante o que é indicado nos passos do tutorial. Paralelamente, é usada a classe QPainter dentro do método paintEvent reimplementado, para pintar um dos QPixmap referidos anteriormente para a área visível de *ilumcanvas*, dando assim *feedback* visual ao utilizador do progresso da sua *iluminura*.

O processo de pintura começa quando o utilizador toca com a caneta na área visível de *ilumcanvas*. Nesse momento é emitido um evento do tipo QTabletEvent que desencadeia a chamada do método tabletEvent de *ilumcanvas*.

O método tabletEvent começa por verificar o tipo de evento recebido. Se o evento for do tipo TabletPress ou TabletRelease, significa que o utilizador tocou ou levantou a caneta da superfície do tablet, respectivamente. O processo de pintura começa efectivamente apenas quando são recebidos eventos do tipo TabletMove. Sempre que é detectado um evento do tipo TabletMove é extraída e armazenada a coordenada contida nesse evento, correspondente à posição relativa da ponta da caneta dentro da área de *ilumcanvas*. Para além desta coordenada, é mantida também a coordenada obtida do evento TabletMove anterior. São necessárias sempre duas coordenadas para colmatar as falhas de continuidade que ocorrem durante a detecção dos traços desenhados pelo utilizador. Por mais sensível que seja a detecção do posicionamento da caneta, é inevitável que as posições encontradas apresentem saltos na continuidade dos traços. Para colmatar esta situação recorreu-se ao armazenamento de duas coordenadas, uma correspondente à posição actual da caneta e outra correspondente à posição anterior. Juntas permitem criar uma linha contínua entre as duas posições e assim compensar a falta de informação que existe em relação ao posicionamento da caneta.

Depois de armazenadas as coordenadas, é chamado o método `updatePen`, responsável pela configuração do objecto `QPen` de acordo com a pressão detectada e com o estado actual das propriedades de `ilumcanvas`, nomeadamente, a cor da tinta escolhida, a quantidade de tinta disponível e as definições da caneta.

Primeiro é configurada a espessura do traço, calculado a partir da pressão detectada e das actuais definições da caneta. De seguida, é configurada a cor, obtida a partir da tinta escolhida pelo utilizador e do nível de opacidade. A opacidade é influenciada tanto pela quantidade de tinta disponível, como pela pressão exercida pela caneta. A quantidade de tinta disponível é uma propriedade de `ilumcanvas` em constante actualização, influenciada tanto pela pressão, como pela dimensão dos traços desenhados. A pressão, por sua vez, é extraída dos eventos recebidos através do método `QTabletEvent::pressure()`.

A definição da opacidade começa com o cálculo da quantidade de tinta necessária para desenhar com a pressão detectada. Devido ao valor da pressão estar compreendido entre 0 e 1, é necessário convertê-lo para um intervalo de valores que permita usá-lo directamente no canal alfa da classe `QColor`. Esta conversão é feita multiplicando o valor da pressão por 255, obtendo assim um valor dentro do intervalo desejado. Após a conversão, é comparado o valor obtido com a quantidade de tinta disponível no momento. Se a quantidade de tinta disponível não for suficiente para satisfazer o pedido, ou seja, se a quantidade de tinta disponível for inferior à necessária, é usado directamente o valor da quantidade de tinta disponível no canal alfa do objecto `QColor`. Por outro lado, se a quantidade de tinta disponível for suficiente para satisfazer o pedido, ou seja, se a quantidade de tinta disponível for superior ou igual à necessária, é usado directamente o valor da quantidade de tinta necessária. No fundo, é escolhido sempre o mínimo dos dois valores para definir o nível de opacidade. Assim, garante-se sempre o nível de opacidade adequado, quer seja pela falta de tinta, quer seja pela pouca pressão exercida pelo utilizador. O nível de opacidade é definido através do método `setAlpha` da classe `QPen`, que recebe como argumento o valor encontrado anteriormente. Finalmente, é chamado o método `setColor` da classe `QPen`, que recebe como argumento o objecto `QColor`, já com o nível de opacidade configurado.

Por fim, o método `updatePen` actualiza a quantidade de tinta disponível, subtraindo a quantidade de tinta que foi efectivamente gasta pelo utilizador. Este valor é dado pelo produto entre a quantidade de tinta gasta na última posição detectada e a dimensão do traço desenhado. A dimensão do traço é dada pelo cálculo da distância de Manhattan entre as duas coordenadas armazenadas. A quantidade de tinta gasta na última posição, por sua vez, é dada pelo produto entre o nível de opacidade encontrado anteriormente e uma constante que permite ajustar a velocidade de consumo da tinta.

Depois de concluída a configuração do objecto `QPen`, são construídas duas instâncias da classe `QPainter` com apontadores para os objectos `QPixmap` mencionados anteriormente, nomeadamente, um com fundo transparente e outro com fundo da cor do pergaminho. De seguida, é chamado o método `paintPixmap`, uma vez para cada um dos objectos `QPainter` criados, que é responsável pela configuração do objecto `QPainter` recebido e pela realização do traço criado pelo utilizador. A configuração consiste essencialmente na definição do objecto

QPen a utilizar e na activação do modo *anti-aliasing* para suavizar os contornos dos traços criados pelos utilizadores. A realização do traço criado, por sua vez, consiste na chamada do método `drawLine` da classe `QPainter`, que pinta efectivamente para o objecto `QPixmap`, uma linha a partir das coordenadas armazenadas. Após concluída a actualização dos objectos `QPixmap`, é chamado o método `update()`, que desencadeia o refrescamento da área visível de `ilumcanvas`, através da chamada do método `paintEvent`. Este método pinta para a área visível de `ilumcanvas` um dos objectos `QPixmap` usados, garantindo assim *feedback* visual ao utilizador do progresso da sua iluminura.

4.1.6 Preparação de tintas

A janela de preparação de tintas é representada pela classe `ilumworkbench`, que estende `QFrame` e é responsável pela organização e funcionamento de todos os *widgets* que compõem a sua interface.

Tal como sucede nas restantes janelas da aplicação, todos os componentes que compõem esta interface são criados e organizados quando a aplicação é inicializada. Contudo, contrariamente ao que acontece na janela de produção de iluminuras e janela informativa, onde todas as informações respeitantes à iluminura seleccionada são recebidas e carregadas quando o utilizador faz a selecção, aqui o carregamento da informação é feito em duas fases distintas. Uma primeira fase, que ocorre após a escolha da iluminura, onde é chamado o método `loadWorkbench` da classe `ilumworkbench`, que recebe e armazena um objecto do tipo `ilumdata` com todas as informações associadas à iluminura escolhida, e uma segunda fase onde é feito efectivamente o carregamento do conteúdo para os respectivos *widgets*, após o utilizador definir qual a tinta que deseja preparar. Este carregamento faseado deve-se sobretudo à dependência que existe em relação às decisões tomadas pelo utilizador. O carregamento depende primeiramente da escolha da iluminura, que restringe imediatamente o lote de tintas disponíveis para preparação, e depende ultimamente da escolha da tinta que o utilizador deseja preparar. O facto de serem necessárias duas escolhas separadas, ao invés de uma como acontece nas outras duas janelas, explica a natureza faseada do carregamento.

A escolha da tinta é feita na janela de produção de iluminuras através do toque num dos botões que se encontram junto aos recipientes das tintas. Quando é seleccionada uma tinta para preparação, é emitido um sinal na classe `ilumworkshop` com o código RGB da tinta escolhida. Este sinal encontra-se ligado a um *slot* da janela principal `VirtualScriptorium`, responsável por desencadear o carregamento do conteúdo para os respectivos *widgets* e pela transição para a janela de preparação de tintas.

Os materiais são representados na interface por instâncias da classe `inklabel`, uma classe que estende `QLabel` para carregar e exibir as imagens dos materiais, ou dos recipientes onde estes se encontram, e que actua como *container* para algumas propriedades associadas aos materiais, nomeadamente, um identificador único, usado durante o arrastamento, e o caminho para uma imagem de uma porção individual do material, usada opcionalmente para representar determinados materiais durante o arrastamento. A aplicação prevê dois tipos de materiais,

aqueles que são usados uma só vez e são arrastados na sua totalidade para dentro do almofariz, e aqueles que são empregados diversas vezes e são arrastados em pequenas quantidades. Enquanto os primeiros necessitam apenas da imagem que os representa e dispensam a imagem opcional, os segundos já requerem duas imagens, a imagem do material ou recipiente e uma imagem de uma porção individual para representação durante o arrastamento.

Como foi referido anteriormente em 3.1.2.3, o tutorial consiste fundamentalmente numa receita composta por uma sequência de passos que o utilizador deve seguir de forma a preparar a tinta seleccionada. O primeiro passo do tutorial, assim como todos os restantes, contém o caminho das imagens com as instruções a seguir, o caminho da imagem com a representação actual do estado do almofariz e ainda a definição dos eventos que desencadeiam as transições automáticas para os próximos passos. A listagem 4.7 apresenta-se um exemplo de um passo do tutorial.

Listing 4.7: Passo da preparação de tintas especificado no ficheiro XML

```
1 <state id="1">
2   <!-- Imagem com informação complementar -->
3   <viewport x="90" y="90">viewport.png</viewport>
4   <!-- Imagem com informação textual -->
5   <image>image.png</image>
6   <!-- Imagem do almofariz -->
7   <mortar>mortar.png</mortar>
8   <!-- Condições de transição -->
9   <triggers>
10    <trigger type="2" state="5">75</trigger>
11    <trigger type="1" state="6">2</trigger>
12  </triggers>
13 </state>
```

O carregamento de passos do tutorial pressupõe essencialmente a redefinição dos eventos que desencadeiam as transições automáticas e o carregamento das imagens para os respectivos *widgets*.

Contrariamente ao que sucede no tutorial de produção de iluminuras, onde existem botões de avanço e recuo, na preparação das tintas as transições são desencadeadas automaticamente, dependendo exclusivamente das acções praticadas pelo utilizador na área de mistura e moagem. Para cada passo do tutorial é possível definir uma série de actividades, que devem ser desempenhadas pelos utilizadores de modo avançar na preparação da tinta escolhida. Quando é detectado que o utilizador actuou de acordo com o estabelecido, é desencadeada uma transição para o passo definido no evento detectado. A única excepção ao conceito de transições automáticas ocorre quando o utilizador, por qualquer razão, deseja reiniciar o processo de preparação. Neste caso o utilizador pode realizar uma transição manual para o primeiro passo do tutorial, através do toque num botão localizado no canto superior esquerdo da interface. O pedido de reinicialização, para além de provocar o carregamento de uma imagem com o almofariz vazio e o carregamento das primeiras instruções do tutorial, conduz também à reposição de todos os materiais necessários à preparação.

A preparação das tintas prevê dois tipos de eventos, eventos de adição de novo material e eventos de moagem e mistura. Os eventos de adição de material definem o identificador único do material que é necessário adicionar ao almofariz para prosseguir com a preparação. A transição para o próximo passo só é concretizada quando é detectado o arrastamento do material, cujo identificador único corresponde ao definido num dos eventos de adição disponíveis. Os eventos de mistura e moagem, por sua vez, definem a percentagem de mistura/moagem do composto que deve ser atingida para desencadear a transição para o passo definido no evento detectado.

Ao considerarmos que a cada passo do tutorial corresponde um estado diferente da interface e uma série de eventos de transição, podemos olhar para a estrutura do tutorial como uma máquina de estados, onde cada passo do tutorial corresponde a um estado diferente e os eventos que desencadeiam as transições podem ser consideradas as condições que conduzem à mudança de estado. Na figura 4.5 apresenta-se um diagrama com a máquina de estados associada à preparação da tinta azul, que é usada na produção de diversas iluminuras e encontra-se disponível na aplicação para preparação.

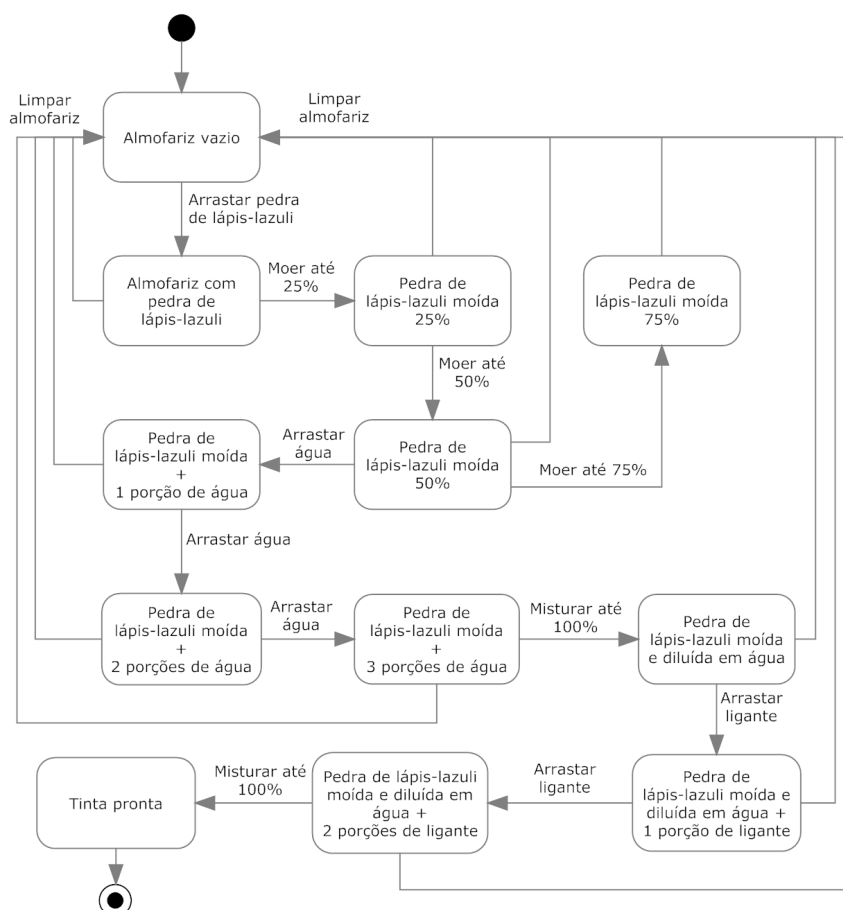


Figura 4.5: Máquina de estados associada à preparação da cor azul

A interface desta janela encontra-se organizada de forma similar à da janela de produção de iluminuras, sendo também composta por três componentes principais.

Um componente responsável pelos materiais, onde estes se encontram prontos a serem arrastados para o almofariz. Este componente é representado na interface pela classe `inkmaterials`, que estende `QFrame` e é responsável pela organização, apresentação e funcionamento de todos os objectos do tipo `inklabel` necessários à preparação da tinta escolhida.

Um componente responsável pelo tutorial de preparação das tintas, similar ao tutorial de produção de iluminuras, sendo também composto por uma instância da classe `QLabel`, que exibe imagens com informações textuais sobre o que o utilizador deve fazer a cada momento, e por um *widget* que permite navegar no seu conteúdo com *kinetic scrolling*, que exibe fotografias capturadas no *workshop* real, com os procedimentos reais que se tentam simular. O desenvolvimento deste último componente pode ser consultado em maior detalhe na secção 4.1.5.

Por fim, um componente destinado à mistura e moagem dos materiais, onde o utilizador pode efectivamente preparar a tinta escolhida. Este componente é representado pela classe `inkmortar`, que estende `QFrame` e é responsável pela exibição, organização e funcionamento dos *widgets* que compõem a sua interface, destacando-se as funcionalidades associadas ao arrastamento dos materiais e à sua mistura/moagem. A interface deste componente é composta por uma instância da classe `QLabel`, que apresenta sempre a imagem actual do almofariz e representa o local para onde os materiais devem ser arrastados, outra instância de `QLabel`, que apresenta um diagrama informativo sobre as transformações que sofrem os materiais durante o processo de mistura/moagem, e uma barra de progresso, representada pela classe `QProgressBar`, que permite aos utilizadores aferir em tempo real a evolução do processo de mistura/moagem.

Como referido anteriormente, o conceito de preparação das tintas assenta sobretudo em dois tipos de interacções: na adição de um novo material ao almofariz e na mistura/moagem do seu conteúdo. Quando é detectada alguma destas acções, são comparadas as suas propriedades com os eventos de transição definidos no estado corrente do tutorial. Ao verificar-se que é cumprida alguma das condições, é desencadeada a transição para o estado definido no evento detectado.

A adição de um novo material ao almofariz consiste essencialmente no arrastamento do material desejado, da área onde este se encontra, para a área de mistura e moagem. Esta funcionalidade foi desenvolvida recorrendo ao já mencionado conceito *drag and drop*, adaptado na janela de selecção de iluminuras para o arrastamento dos *thumbnails*, que compreende o arrastamento de objectos `QObject`, de um *widget* para outro. O desenvolvimento da janela de selecção de iluminuras, assim como o desenvolvimento das funcionalidades base associadas ao conceito *drag and drop*, comuns a estas duas janelas, pode ser consultado pormenorizadamente na secção 4.1.4.

Contrariamente ao que sucede na janela de selecção de iluminuras, onde os *thumbnails* são arrastados dentro do mesmo *widget*, aqui os materiais são efectivamente arrastados de um *widget* para outro. São arrastadas instâncias da classe `inklabel`, do objecto `inkmaterials`, para o objecto `inkmortar`. Esta transferência de objectos entre *widgets* obriga a que os métodos associados a esta funcionalidade se encontrem também separados. Neste caso, o método `mousePressEvent`, que inicia o processo de arrastamento, encontra-se na classe `inkmaterials` e o método

dropEvent, que conclui este processo, encontra-se na classe `inkmortar`.

O método `mousePressEvent` começa por capturar o objecto `inklabel` tocado pela caneta e por criar uma instância da classe `QMimeTypeData` com o identificador único do material capturado. A seguir é criada uma instância da classe `QDrag`, onde é armazenado o objecto `QMimeTypeData` acabado de criar e é definida a imagem que irá representar o material durante o arrastamento. A definição desta imagem depende exclusivamente do tipo de material capturado. Caso o material arrastado só possa ser usado uma vez, é usada a imagem do objecto `inklabel` capturado para representar o material durante o arrastamento. Caso o material arrastado possa ser usado múltiplas vezes, é usado o caminho da imagem opcional, contido no objecto `inklabel` capturado, para criar um novo objecto `QPixmap` com a imagem de uma porção individual do material arrastado. Este objecto `QPixmap` é depois usado para representar o material durante o arrastamento. Por fim, é chamado o método `exec()` do objecto `QDrag`, que inicia o arrastamento e fica bloqueado até obter resposta do método `dropEvent` da classe `inkmortar`.

O método `dropEvent`, por sua vez, extrai do objecto `QDropEvent` recebido o identificador único do material arrastado e verifica se o estado corrente do tutorial contempla algum evento de adição de novo material, cujo identificador corresponda ao recebido. Caso isso se verifique, significa que foi arrastado o material correcto para o almofariz. Segue-se a definição da acção que será retornada em `exec()` no método `mousePressEvent`, que, por sua vez, desencadeará o procedimento adequado para a conclusão da operação. Por fim, é emitido um sinal com o identificador do próximo passo, ligado a um *slot* da classe `ilumworkbench`, responsável por carregar o passo do tutorial correspondente ao identificador recebido.

A mistura/moagem dos materiais presentes no almofariz é feita com a caneta, simulando a utilização do pilão numa situação real de preparação de tintas. O nível de mistura/moagem é influenciado tanto pela distância percorrida pela caneta na área do almofariz, como pela pressão exercida pela ponta da caneta nessa área. Estas características remetem para o desenvolvimento da janela de produção de iluminuras, onde o desenvolvimento das funcionalidades base associadas à pintura e desenho contemplam também a detecção da pressão e o cálculo da distância percorrida pela caneta. O desenvolvimento da janela de produção de iluminuras pode ser consultado pormenorizadamente na secção 4.1.5.

O processo de mistura/moagem começa quando o utilizador toca com a caneta na área visível do almofariz. Nesse momento é emitido um evento do tipo `QTabletEvent` que desencadeia a chamada do método `tabletEvent` da classe `inkmortar`. O método `tabletEvent` começa por verificar o tipo de evento recebido. Se o evento for do tipo `TabletPress` ou `TabletRelease` significa que o utilizador tocou ou levantou a caneta da superfície do tablet, respectivamente. O processo de mistura/moagem começa efectivamente apenas quando são recebidos eventos do tipo `TabletMove`. Sempre que é detectado um evento do tipo `TabletMove` é verificado se o passo corrente do tutorial contém algum evento de mistura/moagem. Caso isso se verifique, é actualizado o nível de mistura/moagem, é actualizada a barra de progresso e é verificado se estão reunidas as condições para desencadear a transição para o próximo passo do tutorial.

O incremento do nível de mistura/moagem é dado pelo produto entre a distância percorrida pela caneta sobre o almofariz e a pressão detectada na superfície do tablet durante essa

interacção. A pressão é obtida directamente do objecto `QTabletEvent` recebido. A distância, por sua vez, é obtida através do cálculo da distância de Manhattan entre a coordenada do evento recebido e a coordenada do evento anterior. Todavia, devido à possibilidade dos resultados obtidos apresentarem valores demasiado elevados, que resultariam na mistura/moagem praticamente imediata do conteúdo do almofariz, definiu-se uma constante, que quando multiplicada pelos resultados obtidos, permite convertê-los para valores consideravelmente mais reduzidos e adequados ao pretendido. Para além disso, a introdução desta constante permite ajustar a sensibilidade e controlar de forma eficaz a velocidade com que os materiais são moídos e misturados.

Por fim, é testado se o nível de mistura/moagem cumpre alguma das condições dos eventos do tutorial. Se o nível de mistura/moagem ultrapassar o nível definido num dos eventos do passo corrente do tutorial, é emitido um sinal com o identificador do próximo passo do tutorial, ligado a um *slot* da classe `ilumworkbench`, responsável pelo seu carregamento.

4.2 Painel interactivo para exploração de iluminuras medievais

O processo de desenvolvimento desta aplicação começou pela produção da estrutura de suporte na biblioteca `openFrameworks` e pelo desenvolvimento dos componentes de suporte à interacção multitoque. Seguiu-se o desenvolvimento dos componentes necessários à representação da informação disponível, assim como o desenvolvimento dos mecanismos indispensáveis à navegação na interface. Por fim, foi desenvolvido o componente responsável pela exibição dos conteúdos criados pelos utilizadores e o componente necessário à exibição dos códigos virtuais. Paralelamente foram sendo criadas as estruturas auxiliares para o armazenamento da informação necessária à aplicação, assim como o componente que permite carregar esses dados para memória e disponibilizá-los aos restantes elementos que compõem a aplicação.

4.2.1 Arquitectura

A aplicação desenvolvida apresenta uma arquitectura adaptada ao funcionamento da plataforma `openFrameworks` (OF) [LW11]. Uma aplicação desenvolvida em OF segue um paradigma de programação orientado a eventos, tendo como base uma classe denominada `ofBaseApp` que define os métodos responsáveis pelo tratamento de eventos a partir dos quais são construídas estas aplicações. Um projecto OF recorre a um método `main` para desencadear a inicialização e arranque da aplicação, assim como a inicialização de outra classe, normalmente denominada `testApp`, que estende a classe `ofBaseApp`, herdando todas as suas propriedades e reimplementando os principais métodos associados ao tratamento dos eventos de uma aplicação interactiva, nomeadamente, métodos para inicializar, desenhar, actualizar e métodos responsáveis pelo tratamento de eventos oriundos dos dispositivos de entrada.

Como já foi referido, a navegação na interface da aplicação encontra-se organizada hierarquicamente numa estrutura em árvore, consequentemente a arquitectura da aplicação, no que concerne aos componentes responsáveis pela visualização, encontra-se organizada de forma

análoga. A arquitectura da aplicação encontra-se organizada numa árvore de objectos, cuja raiz é uma instância da classe `testApp`, onde cada nó é um componente responsável pela gestão, funcionamento e integração dos seus descendentes com a restante aplicação, e as folhas são as janelas responsáveis pela apresentação da informação no ecrã.

Em cerca de 80% dos casos, as folhas da árvore de objectos são constituídas por janelas simples, construídas a partir de imagens, botões e *pop-ups*. Este facto conduziu à criação de uma classe genérica, denominada `Panel`, que permite desenhar para o ecrã um conjunto de botões, imagens e *pop-ups*, assim como apresentar botões de retrocesso e tratar eventos oriundos dos dispositivos de entrada. A classe `Panel` é construída a partir de um conjunto de instâncias das classes `Node`, `Sprite` e `Popup`, que estendem `ofImage` e representam na aplicação botões, imagens e *pop-ups*, respectivamente. A classe `Node` oferece as funcionalidades base para a construção de botões na aplicação, nomeadamente, o carregamento de imagens circulares com pré-visualizações da informação representada, mecanismos associados à captura do toque dos utilizadores e emissão de eventos para os seus antepassados na árvore de objectos. A classe `Node` prevê ainda que o botão possa ser arrastado pela interface, voltando à posição inicial com velocidade decrescente, depois do utilizador o libertar.

Quando é detectado alguma interacção do utilizador com alguma destas janelas, quer seja através do toque num dos botões de selecção, quer seja através do toque num botão de retrocesso, é emitido um evento pelo respectivo objecto `Panel`, que depois é tratado pelo seu pai na árvore de objectos.

Todos os objectos pertencentes a esta árvore, ou seja, todos os nós da árvore de objectos, implementam uma interface, denominada `iPanel`, que define os métodos responsáveis pelo tratamento dos eventos de desenho dos elementos visuais no ecrã e dos eventos multitoque detectados na superfície de interacção. No fundo, trata-se de uma interface com os principais métodos de tratamento de eventos reimplementados pela classe `testApp`, adaptados ao contexto da aplicação.

Listing 4.8: Interface `iPanel`

```
1 void update();  
2 void draw();  
3 void surfaceTouched(int x, int y);  
4 void surfaceSwiped(int x, int y);  
5 void surfaceReleased(int x, int y);
```

O facto de todas as classes presentes na árvore implementarem esta interface permite que todos os eventos recebidos pela classe `testApp` sejam disseminados pela árvore de objectos até chegarem à folha correspondente à janela visível no momento. Ao chegarem à folha, os eventos são tratados de acordo com o seu tipo. Os eventos de refrescamento de imagem provocam o desenho dos elementos visuais da janela em questão no ecrã. Os eventos multitoque, por sua vez, implicam a execução das acções previstas para as interacções detectadas. Este padrão de funcionamento é seguido tanto pela classe `Panel`, como pelas classes associadas às janelas mais complexas da aplicação, como as janelas dos códigos virtuais ou da galeria.

Para além dos componentes associados à visualização e gestão da interface, a aplicação compreende ainda os seguintes componentes:

- Um componente dedicado ao tratamento de eventos multitoque detectados na superfície interactiva usada.
- Um componente responsável pelo carregamento de toda a informação necessária ao funcionamento da aplicação.

O desenvolvimento destes componentes pode ser consultado em maior detalhe na secção 4.2.2 e na secção 4.2.3, respectivamente. A aplicação inclui ainda um mecanismo de *logging* que regista todas as interacções dos utilizadores com a aplicação e que serve de suporte à avaliação do comportamento da aplicação em situações de utilização real.

A figura 4.6 apresenta a arquitectura da aplicação desenvolvida, onde se podem observar os componentes mencionados anteriormente, assim como as relações estabelecidas entre eles.

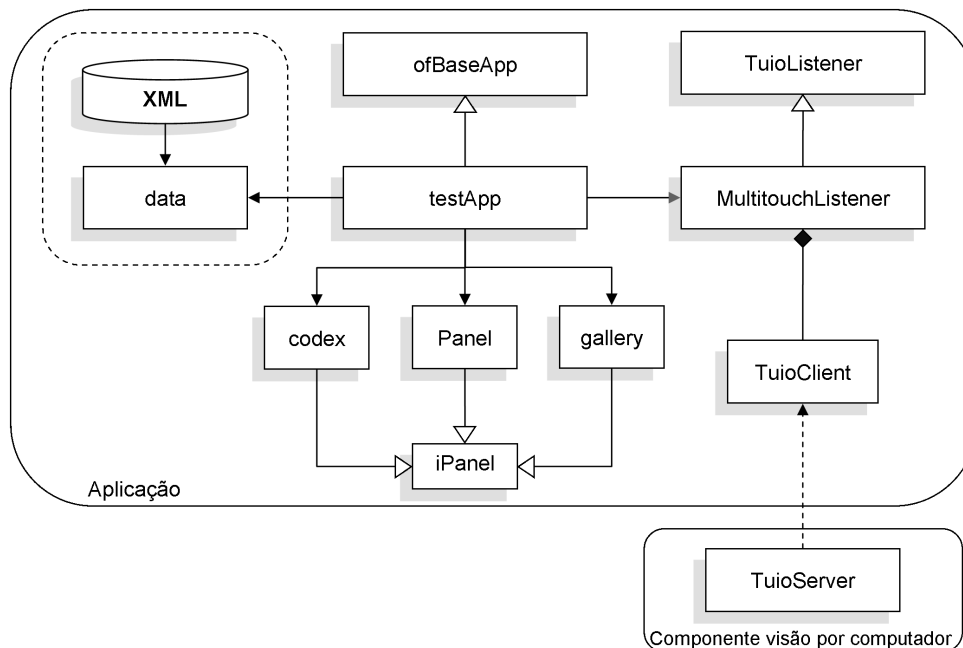


Figura 4.6: Arquitectura do painel interactivo para exploração de iluminuras medievais

A inicialização de todos os componentes apresentados anteriormente é feita aquando do arranque da aplicação no método `setup()` da classe `testApp`, disponibilizado precisamente para a inicialização dos componentes e mecanismos de suporte às aplicações OF. A inicialização é feita de acordo com o código apresentado na listagem 4.9.

Listing 4.9: Inicialização e configuração da aplicação multitoque

```

1  /* Tratamento de eventos multitoque */
2  tuio = new MultitouchListener();
3  ofAddListener(tuio->addTuioCursorEvent, this, &testApp::addTuioCursorHandler);
4  ofAddListener(tuio->removeTuioCursorEvent, this, &testApp::removeTuioCursorHandler);
5  ofAddListener(tuio->updateTuioCursorEvent, this, &testApp::updateTuioCursorHandler);

```

```

6
7  /* Carregar dados a partir dos ficheiros XML */
8  myData = new data;
9
10 /* Criar janelas da aplicação e definir métodos de tratamento de eventos */
11 mainmenu = new Panel(myData->getMMDData());
12 ofAddListener(mainmenu->nodeSelectedEvent, this, &testApp::nodeSelectedHandler);
13 bookOfBirds = new codex(myData->getBookOfBirds());
14 ofAddListener(bookOfBirds->showMainMenuEvent, this, &testApp::showMainMenuHandler);
15 ugcGallery = new gallery(myData->getUgc());
16 ofAddListener(ugcGallery->showMainMenuEvent, this, &testApp::showMainMenuHandler);
17 ...

```

4.2.2 Interacção multitoque

A interacção com a aplicação desenvolvida, como já foi referido, é feita através de uma interface tangível numa superfície interactiva multitoque de grandes dimensões. Deste modo, foi criado um componente dedicado ao tratamento de eventos multitoque, detectados na superfície interactiva da aplicação.

O modelo adoptado para a detecção de eventos multitoque pressupõe uma separação clara entre o componente responsável pela detecção dos eventos multitoque e o componente dedicado ao seu tratamento. A comunicação entre os dois componentes é feita através do protocolo TUIO [KBBC05], que permite precisamente transmitir representações abstractas de superfícies interactivas, incluindo os eventos de toque e o estado de objectos tangíveis, entre os dois componentes já mencionados. O modelo cliente/servidor adoptado permite uma independência total entre detecção e tratamento, tendo como único requisito que a comunicação seja efectuada através do protocolo TUIO. Este conceito pode ser observado em detalhe na figura 4.7.

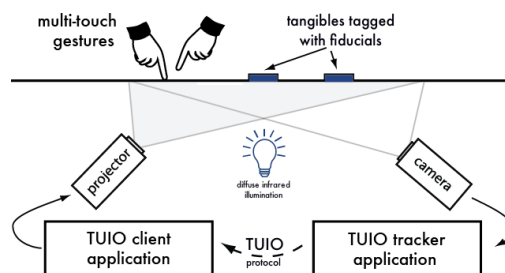


Figura 4.7: Protocolo TUIO

O componente desenvolvido diz respeito apenas ao tratamento de eventos multitoque, ficando a definição do componente responsável pela detecção, aberta a qualquer aplicação que recorra ao protocolo TUIO para comunicar os eventos detectados. Alguns exemplos de bibliotecas capazes de rastrear eventos multitoque de uma superfície interactiva e codificá-los

em mensagens TUIO são os já mencionados Community Core Vision (CCV) [z111], Touchlib [D.11] e reacTIVision [KB07]. Para além destas soluções associadas à tecnologia de infravermelhos, existem ainda diversas aplicações que permitem detectar e reencaminhar eventos resultantes de gestos executados pelo utilizador, para clientes TUIO. Isto permite que a aplicação desenvolvida possa ser utilizada de diferentes formas e em diferentes tipos de *hardware*, desde multitoque em ecrãs capacitivos, até ao reconhecimento de gestos em plataformas como o Kinect.

Este componente é representado na aplicação pela classe `MultitouchListener`, que actua como *listener* de mensagens TUIO em *sockets* TCP/IP e é responsável pela sua descodificação e reencaminhamento para a camada aplicacional. A classe `MultitouchListener` implementa a interface `TuioListener`, que define os métodos dedicados ao tratamento de eventos do protocolo TUIO, apresentados na listagem 4.10.

Listing 4.10: Métodos de tratamento de eventos TUIO

```

1 virtual void addTuioObject (TuioObject *tobj);
2 virtual void removeTuioObject (TuioObject *tobj);
3 virtual void updateTuioObject (TuioObject *tobj);
4 virtual void addTuioCursor (TuioCursor *tcur);
5 virtual void removeTuioCursor (TuioCursor *tcur);
6 virtual void updateTuioCursor (TuioCursor *tcur);
7 virtual void refresh (TuioTime bundleTime);

```

Os métodos definidos pela interface `TuioListener` são chamados pela classe `TuioClient` após a recepção e descodificação das mensagens TUIO presentes no *socket* TCP/IP utilizado. A classe `TuioClient` é o componente central de descodificação do protocolo TUIO, oferecendo uma infraestrutura simples de *callback* baseada na interface `TuioListener`.

Para receber e descodificar mensagens TUIO é necessária a criação de uma instância da classe `TuioClient`, que fica responsável pela recepção, descodificação e interpretação de todas as mensagens recebidas. A seguir são registados nessa instância `TuioClient` objectos que implementem a interface `TuioListener`. Assim, todas as mensagens TUIO recebidas e descodificadas pela instância `TuioClient` originam a difusão de eventos TUIO para os métodos da interface `TuioListener`, implementados pelas classes registadas. No caso da aplicação desenvolvida é registado um objecto do tipo `MultitouchListener`, que implementa a interface `TuioListener`, para tratar os eventos gerados pela instância `TuioClient` criada.

Listing 4.11: Inicialização e configuração de cliente TUIO

```

1 TuioClient *tuioClient = new TuioClient (3333);
2 tuioClient->addTuioListener (this);
3 tuioClient->connect ();

```

Como se pode observar na listagem 4.10, todos os eventos gerados pela classe `TuioClient` recebem como argumento um objecto relativo ao tipo de evento detectado. O protocolo TUIO prevê a transmissão de eventos relativos à manipulação de objectos tangíveis, representados na plataforma por objectos do tipo `TuioObject`, e prevê a transmissão de eventos relativos ao

toque do utilizador na superfície interactiva, representados, por sua vez, por objectos do tipo `TuioCursor`.

Apesar da classe `MultitouchListener` implementar todos os métodos definidos pela interface `TuioListener`, só são utilizados os métodos associados aos objectos do tipo `TuioCursor`, visto que o contexto onde a classe é usada prevê apenas a interacção através do toque do utilizador na superfície interactiva e não contempla qualquer interacção com objectos tangíveis. Os métodos usados dizem respeito à adição, remoção e actualização de cursores TUIO na superfície de interacção, recebendo como argumento objectos `TuioCursor` com a posição e o identificador do cursor detectado. Estes métodos extraem a posição do objecto `TuioCursor` recebido e emitem um evento com essa posição para a classe `testApp`. A classe `testApp`, por sua vez, reencaminha os eventos recebidos para os métodos responsáveis pelo tratamento de eventos multitoque, previstos na interface `iPanel`. Os métodos `addTuioCursor`, `updateTuioCursor` e `removeTuioCursor` da classe `MultitouchListener` desencadeiam a chamada dos métodos `surfaceTouched`, `surfaceSwiped` e `surfaceReleased` da classe `testApp`, respectivamente. Estes métodos, por sua vez, providenciam que todos os eventos recebidos sejam disseminados pela árvore de objectos até chegarem à folha correspondente à janela visível no momento.

4.2.3 Carregamento de dados

Como referido anteriormente em 3.2.1, a informação disponibilizada pela aplicação para consulta e exploração encontra-se armazenada num conjunto de ficheiros de imagens. Estas imagens contêm todos os textos, ilustrações, fotografias e diagramas que constituem a informação semântica associada à aplicação e a partir dos quais é construída a sua interface. A produção destes conteúdos decorreu paralelamente à produção da instalação interactiva e ficou a cargo do Departamento de Conservação e Restauro.

A organização destas imagens é feita de forma estruturada em vários ficheiros XML, com cada ficheiro XML dedicado a um tema diferente. Para que estes ficheiros possam ser carregados e interpretados pela aplicação têm, obrigatoriamente, que respeitar as restrições impostas pelo XML Schema apresentado no anexo E. Este XML Schema estabelece a estrutura e as regras de validação para a definição dos diferentes elementos visuais que compõem uma janela simples da aplicação, nomeadamente, a definição dos botões, imagens, *pop-ups* e acções. Assim, todos os ficheiros XML utilizados para definir as diferentes janelas da aplicação têm de respeitar a estrutura imposta por este XML Schema.

Esta solução, para além de permitir organizar e estruturar os dados disponíveis, permite separar o desenvolvimento da aplicação da produção dos conteúdos e permite evitar que a informação seja *hardcoded*, simplificando substancialmente a integração de novos conteúdos na aplicação. Este método oferece ainda suporte à internacionalização da aplicação, na medida em que é possível criar um ficheiro XML para cada uma das línguas suportadas e carregar apenas aquele que corresponde à língua desejada.

Para além das imagens que servem de base à construção das janelas da aplicação, encontram-se também armazenadas em ficheiros de imagens as iluminuras criadas pelos utilizadores na

aplicação Scriptorium Virtual e os fólhos pertencentes aos códices virtuais disponibilizados pela aplicação. As iluminuras encontram-se armazenadas num repositório de ficheiros dedicado e são carregadas para a galeria quando a aplicação é inicializada. Os fólhos, por sua vez, encontram-se armazenados em repositórios de ficheiros dedicados a cada um dos códices, sendo carregados quando a aplicação é inicializada e surgindo ordenados alfabeticamente na janela dedicada à sua consulta. Para além dos fólhos, cada códice compreende também um ficheiro XML com metadados respeitantes a algumas das suas características (e.g., capa, contracapa e dimensões).

A informação é carregada por um componente específico da aplicação, que faz a análise e interpretação dos ficheiros XML e carrega todas as imagens extraídas dos ficheiros para um conjunto de classes (figura 4.8) dedicado à representação dos diferentes tipos de elementos visuais que compõem a interface da aplicação. Estes objectos são depois distribuídos pelas diferentes janelas da aplicação, onde são utilizados na construção das suas interfaces.

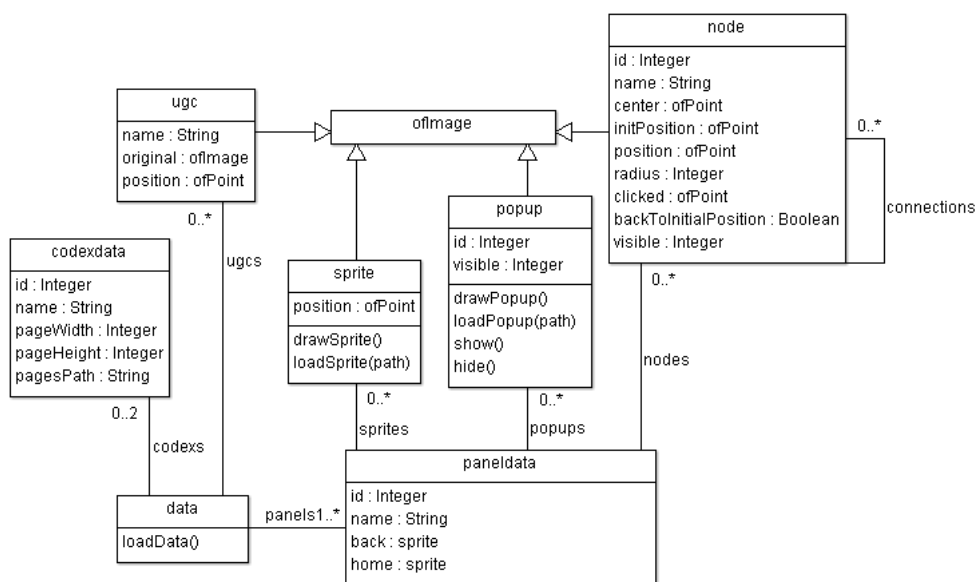


Figura 4.8: Diagrama de classes da aplicação multitoque

4.2.4 Códices virtuais

Um dos principais componentes da aplicação diz respeito às funcionalidades associadas à virtualização de códices medievais. Este componente é responsável pelo carregamento e apresentação dos fólhos de cada códice disponível na aplicação, assim como pelas funcionalidades associadas à sua consulta e exploração.

Os fólhos encontram-se armazenados em ficheiros de imagem, capturados a partir das obras originais. Depois de carregados, os fólhos são ordenados alfabeticamente de acordo com o nome dos seus ficheiros, ficando disponíveis na interface pela mesma ordem da obra original, ou seja, ordenados da capa à contracapa.

A mudança de página é executada pelo utilizador, folheando os fólhos como se de códices

reais se tratassem. Para folhear o códice o utilizador deve pressionar e arrastar com os dedos as pontas inferiores dos fólhos como se estivesse a folhear páginas de um livro. O efeito de folhear apresenta visualmente uma aproximação ao comportamento real das folhas de um livro, permitindo emergir o utilizador ao máximo na consulta das obras e oferecendo um meio de consulta e exploração mais natural aos utilizadores.

A proliferação deste tipo de efeitos em instalações interactivas e dispositivos móveis multi-toque é actualmente uma realidade. O facto de se tentar recriar um efeito sobejamente conhecido pelos utilizadores e o facto deste tipo de efeito se encontrar disseminado em diversos tipos de aplicações, obriga a que o efeito criado seja o mais realista possível de forma a corresponder às expectativas. O conceito adoptado para o efeito de folhear foi inspirado num artigo de Sham Bhangal [Bha11], tendo sido desenvolvido, tal como o resto da aplicação, na plataforma openFrameworks e em OpenGL.

4.2.4.1 Folhear páginas

O conceito apresentado por Sham Bhangal para o desenvolvimento do efeito de folhear assenta na desconstrução deste efeito na perspectiva de encontrar simetrias nas figuras geométricas formadas pelas páginas de um livro real, enquanto são folheadas. Como pode ser observado na figura 4.9, as páginas de um livro ao serem folheadas apresentam sempre uma linha de simetria junto à dobra das folhas, que separa as figuras geométricas simétricas formadas pelas duas páginas reveladas.

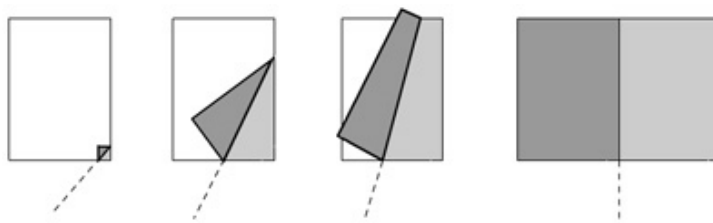


Figura 4.9: Linha de simetria

O virar de página, apesar de constituir um procedimento extremamente comum e simples, ao ser analisado cuidadosamente com vista à sua desconstrução, exhibe alguns comportamentos que tiveram de ser considerados para a reconstrução deste efeito na aplicação desenvolvida.

- A página original (área a na figura 4.10) desaparece lentamente, sendo substituída pela ponta inversa da página, presente no outro lado da folha (área c).
- A página que está por baixo da folha (área b) aparece exactamente à mesma velocidade que desaparece a página original (área a).

O segundo passo na reconstrução deste efeito consiste em decidir qual das três áreas apresentadas (a, b ou c) é que devem ser controladas pelo código desenvolvido. Ao verificar-se que apenas as áreas b e c apresentam simetria, torna-se evidente que será muito mais simples

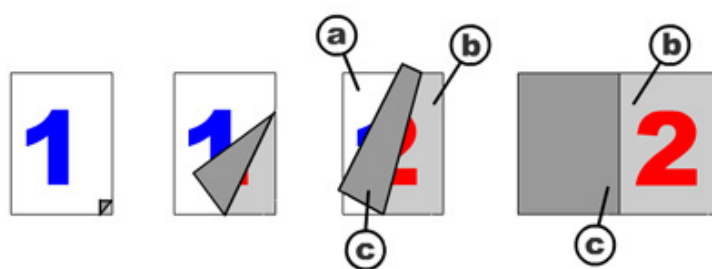


Figura 4.10: As mudanças que ocorrem quando uma página é virada

a representação e controlo destas duas áreas do que da área a. O facto das áreas b e c se relacionarem pela linha de simetria, revela que ambas fazem parte do mesmo processo, tornando as suas animações mais simples de implementar. A área a, por sua vez, não apresenta qualquer tipo de simetria, na realidade apresenta uma forma completamente irregular na maior parte do tempo, tornando a formulação de equações para a sua representação extremamente complexa. Assim, o processo de desenvolvimento é separado em três partes:

- Criar a página direita (área b).
- Criar a página esquerda (área c).
- Completar e construir o efeito.

Estas páginas são criadas fundamentalmente a partir do toque do utilizador na superfície interactiva da aplicação. A interacção do utilizador é iniciada no canto inferior direito ou esquerdo dos livros e continua com o arrastamento dos respectivos cantos na direcção oposta. Sempre que isto sucede é armazenada a abcissa da coordenada relativa ao último toque do utilizador na superfície interactiva. Imediatamente antes da chamada do método `draw()` do `openFrameworks`, é chamado o método `update()` que, por sua vez, chama os métodos responsáveis pela criação das diferentes áreas apresentadas anteriormente a partir da abcissa armazenada.

Depois de construídas as diferentes áreas apresentadas, é chamado o método `draw()` que desenha no ecrã todos os elementos visuais que compõem a interface da aplicação. O conceito adoptado para a construção do efeito de folhear assenta sobretudo na utilização de transparências e sobreposições. Assim, as áreas mencionadas anteriormente são desenhadas por camadas, pela mesma ordem que surgem no livro, ou seja, primeiro é desenhada a página direita (área a), depois é desenhada por cima a futura página esquerda (área c) e finalmente, sobreposta a todas as outras, é desenhada a futura página direita (área b). Inicialmente, as áreas b e c encontram-se ocultas, sendo visível, como seria de esperar, apenas a actual página direita (área a) do livro. Porém, assim que o utilizador começa a folhear a página, começam a surgir as áreas escondidas, recriando em conjunto o efeito desejado.

Para além do arrastamento manual, encontra-se também prevista uma animação automática quando o utilizador não termina de folhear as páginas completamente. Caso o utilizador liberte a folha antes de completar totalmente o movimento, é testado se o canto arrastado se encontra ou não para lá da lombada do livro. No caso de já ter ultrapassado a lombada do livro

o movimento é concluído automaticamente. Caso ainda não tenha ultrapassado a lombada, as folhas retornam automaticamente às posições iniciais. Estas animações automáticas são feitas de forma fluida, recorrendo aos mesmos métodos usados para criar as diferentes áreas durante o arrastamento manual das folhas, mas desta feita sendo chamados automaticamente até se detectar que o efeito terminou.

Apesar da maior parte dos métodos descritos serem relativos ao folhear da página direita, encontra-se também prevista na aplicação o efeito de folhear para as páginas esquerdas. A aplicação permite avançar e recuar na consulta do livro exactamente da mesma forma e com o mesmo efeito. O processo de desenvolvimento é praticamente igual, obrigando apenas a algumas adaptações devido ao diferente posicionamento das páginas e movimentos do utilizador.

Criar a página direita (área b) O primeiro passo na construção da página direita (área b) consiste no cálculo da já mencionada linha de simetria. Ao observarmos as figuras 4.9 e 4.10 podemos verificar que a linha de simetria começa num ângulo de 45° junto ao canto inferior direito do livro e à medida que a página vai sendo folheada, a linha de simetria move-se em direcção à lombada do livro. Quando a linha de simetria atinge a lombada do livro encontra-se num ângulo de 90° , ou seja, na vertical, apresentando uma página totalmente aberta de ambos os lados.

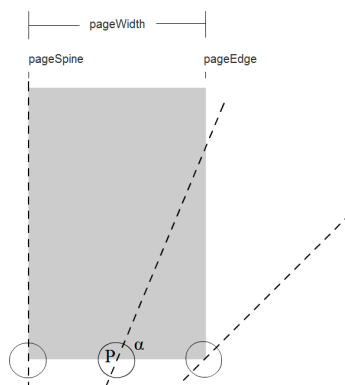


Figura 4.11: Movimento da linha de simetria

Como pode ser observado na figura 4.11, a linha de simetria é definida pelo ponto P e pelo ângulo α . A abcissa do ponto de simetria é dada pela coordenada horizontal correspondente ao toque do utilizador na superfície de interacção ($touchX$), encontrando-se esta interacção limitada à área compreendida entre a borda da página ($pageEdge$) e a lombada do livro ($pageSpine$). A ordenada do ponto de simetria, por sua vez, é dada pela coordenada vertical correspondente à posição em que se encontra o fundo do livro ($pageBottom$).

$$P_{simetria} = (touchX, pageBottom) \quad (4.1)$$

O ângulo da linha de simetria (α) é calculado a partir da abcissa do ponto de simetria ($touchX$) e das posições e dimensões do livro em questão. O ângulo é dado pela seguinte equação:

$$\alpha = 45 \times \left[45 \times \frac{pageEdge - touchX}{pageWidth} \right] \quad (4.2)$$

Esta equação permite obter um ângulo compreendido entre 45° e 90° de acordo com a movimentação da linha de simetria entre a borda da página e a lombada do livro.

Depois de calculada, a linha de simetria é usada na construção da nova página direita (área b). A nova página direita consiste numa textura com a mesma dimensão das restantes páginas do livro e com um posicionamento fixo coincidente com a actual página direita. Todos os *pixels* desta textura que estiverem à direita da linha de simetria são afectados com os valores dos canais RGBA dos *pixels* da página que se pretende revelar. Por sua vez, todos os *pixels* da imagem que estiverem à esquerda da linha de simetria são afectados com valor nulo no canal RGBA dedicado à transparência. Assim, a área à esquerda da linha de simetria encontra-se totalmente transparente, enquanto a área à direita apresenta a imagem normal da página que se pretende revelar.

A textura da página direita é construída a partir de um vector de dimensão igual ao produto entre o número de *pixels* apresentados pela imagem da página direita e o número de posições necessárias para representar cada *pixel* dessa página, ou seja, quatro, correspondentes aos canais RGBA de cada *pixel*.

O preenchimento do vector é feito percorrendo todos os *pixels* da imagem da página direita e verificando individualmente se estes se encontram à esquerda ou direita da linha de simetria. Esta verificação é feita comparando a abcissa da posição de cada *pixel*, com a abcissa do ponto pertencente à linha de simetria, na mesma ordenada da posição do *pixel*. A abcissa do ponto pertencente à linha de simetria é calculada a partir da ordenada da posição do *pixel* e da tangente de α . Caso a abcissa da posição do *pixel* seja inferior à abcissa do ponto pertencente à linha de simetria, as quatro posições do vector correspondentes aos canais RGBA, são preenchidas com os valores 255, 255, 255 e 0, respectivamente. Assim, garante-se a transparência de todos os *pixels* à esquerda da linha de simetria. Por outro lado, caso a abcissa da posição do *pixel* seja superior à abcissa do ponto pertencente à linha de simetria, as quatro posições do vector são preenchidas com os valores dos canais RGBA do *pixel* em questão. Assim, garante-se que todos os *pixels* à direita da linha de simetria mantêm os mesmos valores dos *pixels* da página.

Depois de preenchido, o vector é carregado para a textura, sendo depois desenhada por cima das outras duas páginas. O facto da área à esquerda da linha de simetria apresentar-se totalmente transparente permite que quando esta textura for desenhada no ecrã em último lugar, todas as imagens que foram desenhadas anteriormente no mesmo local possam ser visíveis na interface, contribuindo assim para a construção do efeito pretendido.

Criar a página esquerda (área c) Contrariamente à criação da página direita (área b), que implica apenas a definição dos *pixels* da textura de acordo com a linha de simetria e apresenta um posicionamento fixo, no caso da criação da página esquerda (área c), implica a criação de uma moldura, onde é inserida a imagem da página, com um posicionamento variável, definido pelo toque do utilizador na superfície de interacção.

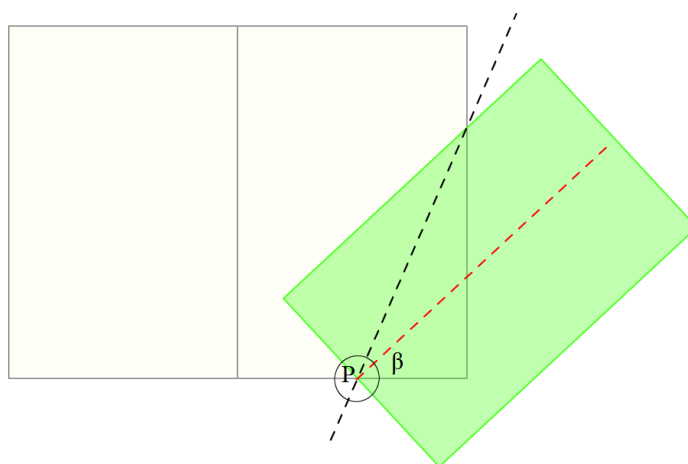


Figura 4.12: Moldura da área c

O posicionamento da moldura é calculado a partir de uma linha semelhante à linha de simetria abordada anteriormente. Esta linha (linha vermelha na figura 4.12) é definida pelo mesmo ponto P , mas desta feita por um ângulo (β) a variar entre 0° e 90° . Como pode ser observado na figura 4.12, a linha vermelha apresenta um ângulo de 0° quando a abcissa do ponto coincide com a borda do livro e a um ângulo de 90° quando a abcissa do ponto coincide com a lombada do livro.

O ângulo da linha vermelha (β) é calculado a partir da abcissa do ponto P (touchX) e das posições e dimensões do livro em questão. O ângulo é dado pela seguinte equação:

$$\beta = \left[90 \times \frac{\text{pageEdge} - \text{touchX}}{\text{pageWidth}} \right] \quad (4.3)$$

Esta equação permite obter um ângulo compreendido entre 0° e 90° de acordo com a movimentação da linha vermelha entre a borda da página e a lombada do livro.

A moldura é definida por quatro pontos, calculados a partir do ângulo β e do ponto P . Primeiro são calculados os dois pontos mais próximos do ponto P , dividindo a área da moldura em triângulos rectângulos a partir deste ponto e calculando trigonometricamente a dimensão dos catetos correspondentes aos desvios entre os dois pontos mais próximos e o ponto P . Por sua vez, os dois pontos mais afastados do ponto P são calculados de modo similar, mas desta feita os cálculos dizem respeito aos desvios entre os dois pontos mais afastados e os dois pontos calculados anteriormente.

A moldura, como pode ser observado na figura 4.12, encontra-se inicialmente escondida, fora da área visível do livro. Assim que o utilizador interage com o canto inferior direito da página e arrasta-o na direcção oposta, é alterada a abcissa do ponto P , permitindo que na próxima actualização do ecrã seja recalculado o ângulo β e os quatro pontos que definem a moldura. Isto resulta na movimentação da moldura e da página nela inserida, recriando o surgimento da ponta inversa da página actual, presente no outro lado da folha.

Completar o efeito O conceito adoptado, apesar de permitir recriar de forma eficaz o efeito pretendido, apresenta algumas limitações relacionadas precisamente com o facto de esta aproximação assentar sobretudo na manipulação de imagens no espaço bidimensional. O desenvolvimento do efeito de folhear baseado na manipulação de imagens diminui substancialmente o realismo pretendido, na medida em que as folhas aparentam ser demasiado rígidas, não apresentando qualquer volume ou relevo e exibindo uma dobra perfeita nas páginas folheadas.

Para tornar o efeito mais convincente aplicaram-se algumas sombras em determinados locais dos livros, à medida que as páginas são folheadas. Esta aproximação compreende uma sombra aplicada ao longo da lombada do livro, criando a ilusão de relevo junto à junção das páginas, assim como uma sombra aplicada na dobra da página folheada, criando a ilusão de volume e flexibilidade das folhas.

A sombra aplicada ao longo da lombada do livro é estática, criada inicialmente de acordo com as dimensões do livro em questão, consiste essencialmente numa textura transparente com um gradiente preto sobreposto à lombada do livro.

Em relação à sombra aplicada na dobra da página, é criada dinamicamente de acordo com o posicionamento da dobra, sendo aplicada directamente na textura da página direita (área b), no momento em que é verificado o posicionamento dos *pixels* em relação à linha de simetria. É aplicado um gradiente preto do lado esquerdo da linha de simetria, ou seja, no lado transparente da textura, começando numa cor preta sólida, junto à linha de simetria, e à medida que a sombra vai-se afastando desta linha adquire uma tonalidade cada vez mais transparente até atingir transparência total. A dimensão e tonalidade desta sombra depende do ângulo α e é aplicada de forma paralela à linha de simetria. É ainda criada uma sombra do lado direito da linha de simetria, ou seja, do lado em que a textura apresenta a imagem da página direita, multiplicando os valores dos canais RGBA respeitantes aos *pixels* da imagem da página direita, contidos no espaço definido para a sombra, por uma variável F com valores compreendidos entre 0.5 e 1. Este intervalo de valores permite escurecer gradualmente a tonalidade das cores da imagem e criar a ilusão de sombra na página direita (área b). O valor desta variável é alterado de acordo com o ângulo α e é dado pela seguinte equação:

$$F = \frac{\alpha}{90} \quad (4.4)$$

Apesar de se tratar de sombras aproximadas, a simples adição destas duas sombras ao efeito previamente apresentado permite transmitir uma verdadeira sensação de volume e oferecer ao efeito uma dose considerável de realismo.

5

Avaliação

Neste capítulo apresentam-se e discutem-se os resultados obtidos nas avaliações feitas às aplicações desenvolvidas no âmbito desta dissertação. O capítulo encontra-se organizado em duas secções, primeiro é apresentada a avaliação da aplicação Scriptorium Virtual e depois a avaliação da aplicação multitoque para exploração das iluminuras medievais.

5.1 Scriptorium Virtual

A apresentação e discussão dos resultados obtidos na avaliação desta aplicação encontram-se divididas em duas secções, correspondentes aos dois estudos efectuados. A primeira secção diz respeito a uma avaliação preliminar, onde foram observados os comportamentos e as interacções dos utilizadores com a aplicação implementada. Esta avaliação foi dedicada exclusivamente a alguns peritos do Departamento de Conservação e Restauro ligados ao projecto e teve como principal objectivo aferir a viabilidade das interfaces e mecanismos de interacção. Na segunda secção são apresentados e discutidos os resultados relativos a uma avaliação da aplicação numa situação de utilização real, junto do seu público-alvo, onde são analisadas as apreciações dos utilizadores ao nível da usabilidade da aplicação e satisfação com a experiência oferecida.

5.1.1 Observação de utilizadores

Esta avaliação teve lugar no dia 15 de Abril de 2011 num laboratório do Edifício Departamental do Campus de Caparica e contou com a participação de alguns membros do Departamento de Conservação e Restauro (DCR) ligados ao projecto. Esta avaliação preliminar teve como principal objectivo aferir a viabilidade da caneta como meio de interacção, avaliar a usabilidade da

interface e a integração dos conteúdos na aplicação. A abordagem adoptada para esta avaliação passou pela observação das interações dos utilizadores com a aplicação, de modo analisar as dificuldades encontradas e o comportamento dos utilizadores durante a utilização.

A avaliação decorreu paralelamente ao *workshop* real de pintura de iluminuras. Os participantes começavam por desenhar e pintar uma iluminura, preparando tintas e desenhando e pintando com os instrumentos e materiais reais. Após concluírem a sua iluminura, os utilizadores recebiam o tablet de modo a reproduzirem novamente a mesma iluminura, mas desta feita na aplicação desenvolvida. Os utilizadores tiveram oportunidade de pintar uma iluminura, preparar tintas e consultar informações, sem interferências, com liberdade para interagirem e explorarem a aplicação como desejassem. Foi usado nesta avaliação o mesmo tablet usado durante o desenvolvimento da aplicação, ou seja, um HP Compaq tc4200 Tablet PC.

Para sistematizar as observações, elaborou-se um formulário que ia sendo preenchido à medida que os utilizadores interagiam com a aplicação. O formulário de observação encontra-se dividido em duas partes. Uma parte que visa traçar o perfil do utilizador e outra parte relacionada com as funcionalidades oferecidas pela aplicação e a satisfação dos utilizadores em relação à experiência oferecida. A versão completa deste formulário, assim como os resultados obtidos durante a observação, podem ser consultados detalhadamente no anexo C e no anexo D, respectivamente.

Em geral, a maioria dos utilizadores manifestou o seu agrado em relação à aplicação, com os mecanismos envolvidos na preparação das tintas e algumas particularidades associadas à produção de iluminuras em principal destaque. A análise dos resultados obtidos durante a observação permitem concluir que os utilizadores consideraram, globalmente, que a aplicação é rápida de aprender e relativamente fácil de utilizar e dominar. Porém, foram também identificadas algumas dificuldades, que viriam a ser confirmadas e sublinhadas dias mais tarde, aquando da segunda avaliação da aplicação, abordada mais à frente. Para além de algumas questões relacionadas com a produção de conteúdos, que se encontram sob a alçada dos responsáveis do DCR, foram identificadas algumas dificuldades de navegação na interface e na reprodução de determinados detalhes das iluminuras com a caneta usada na avaliação.

Relativamente à navegação na interface, determinados utilizadores depararam-se com algumas dificuldades, sobretudo, na identificação de algumas funcionalidades oferecidas pela aplicação, como o arrastamento das imagens com *kinetic scrolling* nos tutoriais e o significado de alguns botões da interface.

Em relação às dificuldades introduzidas pela caneta, alguns utilizadores indicaram que a sensibilidade à pressão teria de ser ajustada para alguns passos do tutorial e foram unânimes que o modelo usado na avaliação apresentava uma ponta demasiado grossa para o pretendido, pois ocultava a área de desenho e dificultava a reprodução dos elementos decorativos de menores dimensões. Para além disso, foi apontada a reduzida dimensão das iluminuras como uma desvantagem, sobretudo quando era necessário reproduzir alguns dos pormenores mais subtils das iluminuras. Como solução, alguns utilizadores sugeriram simplesmente mudar de caneta ou aumentar as dimensões das iluminuras. Relativamente à primeira sugestão, quando a instalação interactiva for definitivamente implementada numa situação de utilização

real, serão utilizados modelos de tablet e caneta diferentes, mais recentes, que poderão reduzir consideravelmente este problema. Em relação à sugestão alternativa, visto que o principal requisito do sistema é divulgar informação sobre o tema, aumentar a dimensão das iluminuras daria uma impressão errada aos utilizadores. A apresentação das iluminuras nas suas dimensões originais é fundamental para divulgar correctamente essa informação. Alternativamente, foi sugerido adicionar uma opção de *zoom*, permitindo assim manter as dimensões originais da iluminura e simultaneamente auxiliar na reprodução dos elementos decorativos mais complexos. Contudo, visto que outro dos requisitos da aplicação passa por capturar parte das dificuldades subjacentes à produção de iluminuras e sendo uma delas motivada precisamente pelas reduzidas dimensões que a maioria das iluminuras apresenta, percebe-se que esta solução gere alguma discussão e não tenha sido imediatamente adoptada. A adopção desta solução encontra-se dependente das características do tablet e da caneta utilizada na versão final da instalação interactiva. Estas dificuldades são exploradas novamente mais adiante, aquando da apresentação da segunda avaliação.

Foi sugerido ainda por alguns utilizadores, adicionar uma área de rascunho onde o utilizador pudesse testar a configuração da caneta antes de pintar directamente na iluminura e adicionar áudio ao tutorial, permitindo ao utilizador desenhar e pintar sem desviar o olhar da área de desenho.

Apesar do reduzido número de participantes, esta avaliação revelou-se importante para o desenvolvimento da aplicação, pois permitiu que esta fosse testada por especialistas na área das iluminuras medievais, os utilizadores mais habilitados para avaliar a aplicação do ponto de vista da adaptação e correcção dos conteúdos apresentados e da fidelidade das operações que se pretendem capturar.

5.1.2 Inquérito aos utilizadores

Esta avaliação foi realizada no dia 29 de Abril de 2011 no âmbito da EXPO FCT 2011, uma mostra de ciência, tecnologia e inovação na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT/UNL) no Campus de Caparica, envolvendo todos os sectores da faculdade nas várias vertentes científicas e tecnológicas. Esta iniciativa, que contou com a presença de mais de 8000 visitantes na sua edição anterior, tem como objectivo divulgar todos os exemplos de inovação tecnológica desenvolvidos na faculdade e nos centros de investigação científica que a faculdade aloja, e aproximar a realidade universitária, na sua vertente de investigação, inovação e tecnologia, com a população, em particular com os alunos do ensino secundário. A avaliação decorreu entre as 9:30h e as 15:30h no Edifício Departamental do Campus de Caparica, onde decorriam as actividades organizadas pelo Departamento de Conservação e Restauro, tendo sido inserida no contexto do *workshop* de pintura de iluminuras. Assim, esta iniciativa reuniu as condições ideais para testar a aplicação desenvolvida numa situação de utilização real, junto do seu público-alvo, existindo inclusive a possibilidade dos utilizadores experimentarem, quer a versão do *workshop* implementada na aplicação, quer o *workshop* real, capacitando-os assim para fazerem uma avaliação comparativa entre o real e o digital.

Foi usado nesta avaliação o mesmo tablet usado durante o desenvolvimento da aplicação, ou seja, um HP Compaq tc4200 Tablet PC, que era entregue aos utilizadores no início da sua participação. Foi dada total liberdade aos utilizadores para interagirem e explorarem a aplicação como desejassem. Os utilizadores tiveram oportunidade de pintar uma iluminura, preparar tintas e consultar informações, sem quaisquer interferências, excepto quando era solicitada alguma informação adicional ou era pedido algum esclarecimento. No fim de cada participação foi pedido aos utilizadores que respondessem a um pequeno questionário, com vista à recolha das suas opiniões em relação à aplicação, de modo a aferir a sua receptividade e usabilidade. O questionário entregue aos utilizadores encontra-se dividido em duas partes. Uma parte que visa traçar o perfil do utilizador para contextualizar a sua experiência com a aplicação e outra parte dedicada à recolha da sua apreciação global, sobretudo relativamente à usabilidade da aplicação e à sua satisfação com a experiência oferecida. A versão completa deste questionário pode ser consultada no anexo [A](#).

5.1.2.1 Apresentação e análise de resultados

Devido ao reduzido espaço de tempo disponível para realizar esta avaliação, visto que a EXPO FCT dura apenas um dia, aliado ao facto da interacção com a aplicação implicar normalmente uma participação demorada, com alguns utilizadores a demorarem mais de 40 minutos, levou a que só fosse possível registar 14 participações. Porém, deste universo de participantes, todos concordaram em responder ao questionário, permitindo assim fazer uma avaliação substancial da aplicação desenvolvida. Os resultados deste inquérito podem ser consultados detalhadamente no anexo [B](#).

Dos inquiridos, 50% são do sexo feminino e 50% são do sexo masculino. Em relação à idade, devido à natureza da EXPO FCT, que tem como público-alvo, sobretudo, alunos do ensino secundário, compreende-se que cerca de 71,4% dos inquiridos pertença à faixa etária dos 10 aos 19 anos. Para além disso, registaram-se cerca de 21,4% de inquiridos na faixa etária dos 20 aos 29 e ainda cerca de 7,2% de inquiridos na faixa etária dos 30 aos 39.

Outro aspecto relevado pelo questionário e que auxilia a traçar o perfil dos utilizadores, diz respeito ao grau de instrução dos inquiridos. Visto que a maioria dos inquiridos são muito jovens, com idades compreendidas entre os 10 e os 19 anos, é natural que cerca de 57,2% dos inquiridos possua o ensino secundário e cerca de 14,3% possua apenas o 3º ciclo do ensino básico. Os restantes inquiridos afirmaram possuir um grau de instrução superior ao ensino secundário, com cerca de 7,1% a possuir um mestrado ou uma pós-graduação, cerca de 7,1% a possuir uma licenciatura e 14,3% com frequência universitária.

Finalmente, para aferir até que ponto é que a familiarização dos inquiridos com as novas tecnologias da informação pode ou não influenciar as suas experiências com a aplicação e em última análise as suas opiniões, foi-lhes perguntado se de facto encontram-se ou não familiarizados com as novas tecnologias da informação e ainda quais os seus hábitos de utilização. Todos os inquiridos mostraram-se familiarizados com as novas tecnologias da informação e todos afirmaram usar diariamente o computador, a internet e o telemóvel. Por outro lado, cerca

de 64,3% dos inquiridos afirmou raramente usar uma consola de jogos.

Em relação à apreciação global dos utilizadores sobre a sua interacção com a aplicação, foi pedido aos inquiridos que classificassem seis afirmações de acordo com o seu nível de concordância, com a classificação 1 a significar «discordo» e a classificação 5 a significar «concordo». A figura 5.1 apresenta um gráfico com os resultados obtidos, assim como as afirmações que foram classificadas pelos utilizadores.

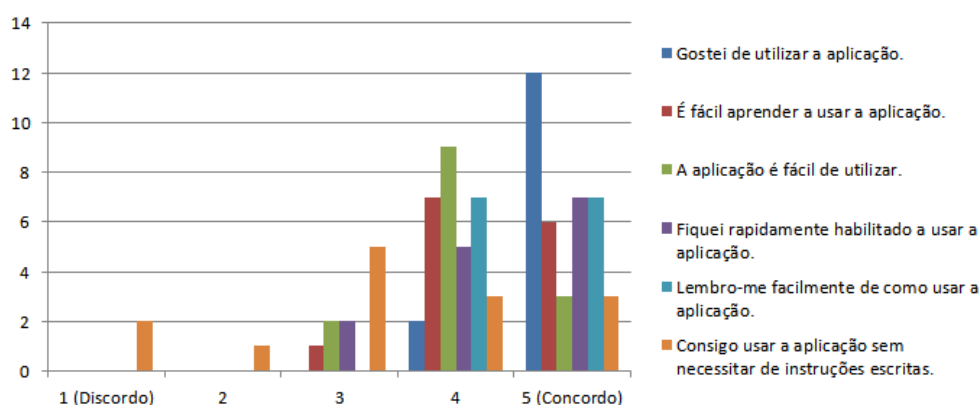


Figura 5.1: Resultados relativos à facilidade de aprendizagem e usabilidade da aplicação

Depois de uma análise cuidadosa do gráfico apresentado, pode-se considerar que, globalmente, a aplicação foi bem aceite pelos utilizadores, com cerca de 85,7% a concordar totalmente (classificação 5) com a primeira afirmação e os restantes 14,3% a concordar parcialmente (classificação 4).

Relativamente à facilidade de aprendizagem e de utilização, os inquiridos dividiram-se entre a concordância total e parcial. A afirmação respeitante à facilidade de aprendizagem apresenta um maior equilíbrio, com 50% dos inquiridos a concordar parcialmente e cerca de 42,9% a concordar totalmente. A afirmação respeitante à facilidade de utilização, por sua vez, apresenta uma maior concordância parcial, correspondente a cerca 64,3% dos inquiridos, e ainda cerca de 21,4% a concordar totalmente com a afirmação. Apenas cerca de 7,1% e 14,3% dos inquiridos adoptou uma postura neutra (classificação 3) relativamente a estas duas afirmações, respectivamente. Assim, pode-se considerar o saldo francamente positivo, com os utilizadores a demonstrarem que não tiveram grandes dificuldades em aprender e em usar a aplicação.

Outro aspecto importante, que diz mais respeito à temática retratada do que propriamente à interface e aos mecanismos de interacção, encontra-se relacionado com a compreensão global do conceito da aplicação e, em determinadas situações, com alguns termos usados. Foi pedido aos inquiridos para classificarem se conseguem usar a aplicação sem o auxílio de instruções escritas. Apesar de cerca de 42,9% dos inquiridos afirmar que concorda totalmente ou parcialmente da afirmação, alguns utilizadores revelaram que o acompanhamento de algumas instruções escritas podia de facto facilitar a utilização da aplicação. Cerca de 14,3% dos inquiridos discordou completamente da afirmação, cerca de 7,1% discordou parcialmente e a maioria, cerca de 35,7%, adoptou uma postura neutra em relação ao tema. Constatou-se que esta discordância partiu, sobretudo, dos poucos utilizadores que ainda não tinham participado

no *workshop* real, reforçando a ideia que a origem do problema diz mais respeito aos conteúdos do que propriamente ao desenho da interface.

Para avaliar a usabilidade da aplicação, foram também apresentadas aos inquiridos algumas afirmações para estes classificarem, respeitantes ao comportamento da interface e à viabilidade da caneta como dispositivo de interacção. A figura 5.2 apresenta um gráfico com os resultados obtidos, assim como as afirmações apresentadas aos utilizadores para classificação.

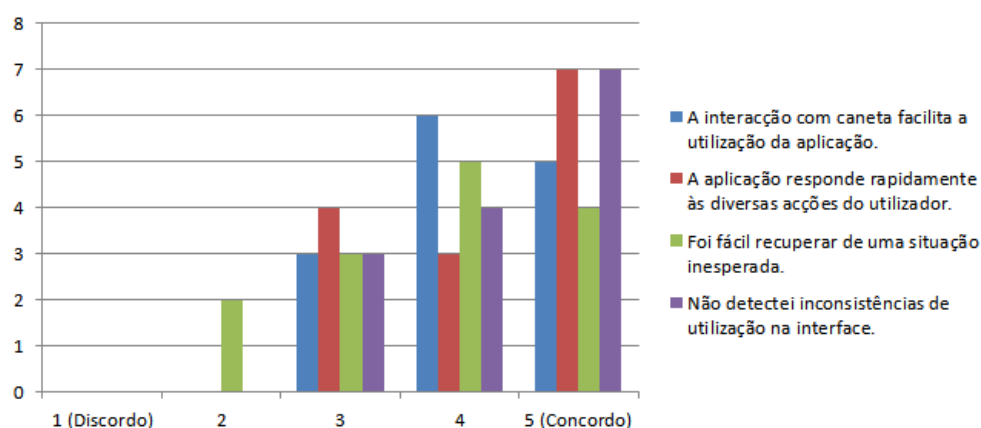


Figura 5.2: Resultados relativos à interacção dos utilizadores com a aplicação

Uma análise deste gráfico revela que a maioria dos inquiridos considera que de facto a caneta facilita a utilização da aplicação, com cerca de 42,9% a concordar parcialmente com esta afirmação e cerca de 35,7% a concordar totalmente. Os restantes inquiridos, cerca de 21,4%, decidiu adoptar uma postura neutra em relação ao assunto. Relativamente ao desenho e à usabilidade da interface, o saldo também é positivo, com os inquiridos na sua maioria a distribuírem-se entre a neutralidade (classificação 3) e a concordância total (classificação 5). Os resultados revelam que 50% dos inquiridos não detectou quaisquer inconsistências na interface e considerou que o tempo de resposta da aplicação é reduzido, os restantes, por sua vez, distribuíram-se entre a neutralidade e a concordância parcial. Globalmente os inquiridos consideram que foi fácil recuperar de uma situação inesperada, contudo, cerca de 14,3% discordou parcialmente da afirmação, demonstrando que sentiram algumas dificuldades com a aplicação. Esta foi, aliás, a única afirmação que registou alguma discordância por parte dos inquiridos, demonstrando a satisfação global dos utilizadores em relação à usabilidade da aplicação e da caneta como meio de interacção.

De seguida foi pedido aos inquiridos para classificarem em termos de facilidade de execução, um conjunto de operações e funcionalidades inerentes à aplicação. A figura 5.3 apresenta um gráfico com os resultados obtidos, assim como as operações e funcionalidades apresentadas aos utilizadores para classificação.

Como se pode verificar, globalmente os inquiridos consideram que a aplicação é de fácil utilização, com a maioria das operações e funcionalidades a serem classificadas entre um grau intermédio e o grau máximo de facilidade, verificando-se uma maior ocorrência da classificação 5 e 4, respectivamente. De facto, apenas algumas operações relacionadas com a caneta,

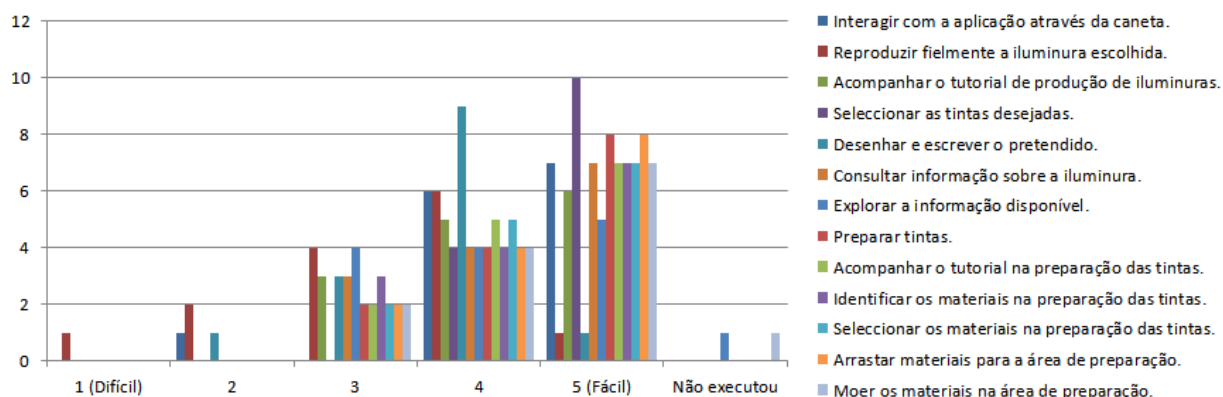


Figura 5.3: Resultados relativos à facilidade de execução de determinadas operações

sobretudo no que diz respeito ao desenho e pintura dos diferentes elementos decorativos que compõem as iluminuras, é que tiveram algumas classificações negativas. Relativamente à reprodução fiel das iluminuras, cerca de 7,1% dos inquiridos considera a operação difícil (classificação 1) e cerca de 14,3% considera-a relativamente difícil (classificação 2), destacando-se neste aspecto a reprodução dos textos, um dos procedimentos mais complexos associados a esta operação. Apesar do reduzido número de inquiridos a considerar a reprodução fiel das iluminuras como difícil, muitos utilizadores admitiram sentir algumas dificuldades em desenhar correctamente determinados detalhes das iluminuras, quer durante as interações, quer no campo do inquérito destinado às sugestões e comentários.

Destaque ainda para o facto de praticamente todos os inquiridos terem explorado por completo a aplicação. Apenas cerca de 7,1% dos inquiridos não explorou as informações disponíveis sobre a iluminura e admitiu não ter moído e misturado os materiais durante a preparação das tintas.

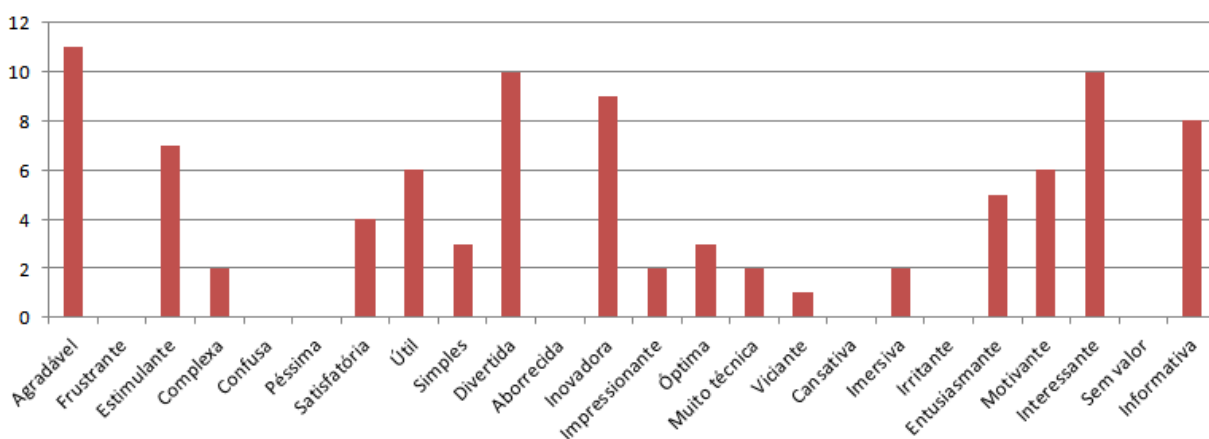


Figura 5.4: Resultados relativos ao envolvimento emocional dos utilizadores com a aplicação

A seguir foi apresentado aos inquiridos um conjunto de expressões para que estes pudessem escolher aquelas que melhor expressassem a sua opinião em relação à aplicação acabada

de usar. A figura 5.4 apresenta um gráfico com os resultados obtidos, assim como as expressões apresentadas aos utilizadores. Destaque para as expressões agradável, divertida, interessante, inovadora e informativa, que reuniram a preferência de mais de metade dos inquiridos. Relativamente às expressões menos positivas, apenas cerca de 14,3% dos inquiridos considerou a aplicação complexa e demasiado técnica, devido às dificuldades já mencionadas em reproduzir com a caneta alguns detalhes das iluminuras.

Por fim, foi pedido aos inquiridos que deixassem alguns comentários e sugestões relativamente ao sistema desenvolvido. Cerca de 42,9% dos inquiridos respondeu ao desafio, deixando comentários e sugestões construtivas. Apesar da recepção francamente positiva, com os comentários a transparecer a satisfação global dos utilizadores em relação à aplicação, a maioria dos comentários refere também algumas dificuldades sentidas pelos inquiridos durante a sua interacção com o sistema.

A questão mais abordada, já mencionada anteriormente, diz respeito à dificuldade em reproduzir determinados detalhes das iluminuras. A maior parte dos inquiridos apontou a caneta usada como o principal obstáculo, com alguns a referir que a ponta da caneta é demasiado grossa para o pretendido e outros a perguntar se não seria possível alterar a ponta da caneta para uma mais fina. Apesar de a caneta permitir traços extremamente finos, a elevada espessura apresentada pela sua ponta diminui de facto a precisão dos traços produzidos, sobretudo porque oculta ligeiramente a área de desenho, dando azo a alguns erros e imperfeições, acabando por influenciar substancialmente a qualidade final das iluminuras. Como solução, alguns utilizadores sugeriram eliminar a raiz do problema e simplesmente utilizar uma nova caneta, menos grossa, conjuntamente com um novo tablet. Por outro lado, alguns utilizadores sugeriram uma melhor adaptação ao *hardware* disponível, aumentando as dimensões das iluminuras ou adicionando uma opção de *zoom* para facilitar a reprodução dos elementos decorativos de dimensões mais reduzidas.

Alguns utilizadores mencionaram também uma ligeira discrepância existente entre a posição da ponta da caneta na superfície do tablet e o local onde surgem de facto os elementos desenhados. Devido ao modelo do tablet usado durante a avaliação apresentar um espaço considerável entre a superfície responsável pela detecção do posicionamento da caneta e o ecrã propriamente dito, quanto maior for o ângulo de visão do utilizador em relação ao ecrã do tablet, maior será esta discrepância. Alguns utilizadores referiram inclusive que o problema se acentuava à medida que a ponta da caneta se aproximava da margem superior do tablet, precisamente porque o ângulo de visão aumenta proporcionalmente. Este problema manifestou-se sobretudo durante a fase inicial da avaliação, numa altura em que o tablet se encontrava pousado numa mesa e apresentava um ângulo de visão extremamente elevado. À medida que a avaliação foi decorrendo foram testadas diferentes posições, terminando com os utilizadores a colocar o tablet ao colo e assim desfrutar de uma maior liberdade de movimentos para desenhar e de um ângulo de visão que elimina completamente quaisquer discrepâncias. Para o modelo de tablet usado durante a avaliação, a solução passou de facto por posicioná-lo perpendicularmente à visão do utilizador, contudo, quando a instalação interactiva for definitivamente implementada numa situação de utilização real, serão utilizados modelos de tablets

mais recentes e este problema poderá ser reduzido.

Alguns utilizadores, sobretudo aqueles que tiveram oportunidade de participar anteriormente no *workshop* real, comentaram que o facto da superfície do tablet não apresentar qualquer atrito, contrariamente ao que sucede quando se desenha em pergaminho ou mesmo em papel, diminui de certa maneira a precisão necessária para desenhar e pintar. Alguns utilizadores comentaram ainda que a caneta apresentava pouca sensibilidade, enquanto outros comentaram precisamente o contrário, ou seja, que a caneta apresentava demasiada sensibilidade.

Para além dos comentários e sugestões que os utilizadores escreveram no questionário, foram sendo registados todos comentários que os utilizadores iam proferindo durante a sua interacção com a aplicação. Para além das questões já abordadas, os comentários focaram-se sobretudo no desenho da interface e na produção de conteúdos, com alguns utilizadores a não perceberem imediatamente algumas funcionalidades menos óbvias da aplicação, como o arrastamento, mistura e moagem dos materiais na preparação das tintas.

5.2 Painel interactivo para exploração de iluminuras medievais

A avaliação desta aplicação, que decorreu no dia 17 de Setembro de 2011 e contou com a participação de 9 utilizadores, teve como objectivo, sobretudo, aferir a eficácia dos meios de interacção usados e do desenho da interface no que diz respeito à navegação, exploração e consulta dos diferentes tipos de informação veiculados.

Os utilizadores tiveram acesso à aplicação sem quaisquer interferências, podendo interagir e explorar a aplicação como desejassem. No fim de cada interacção foi pedido aos utilizadores que respondessem a um pequeno questionário, com vista à recolha das suas opiniões relativamente à experiência oferecida pela aplicação, quer do ponto de vista da sua usabilidade, quer do ponto de vista da sua receptividade junto do público.

O questionário entregue aos utilizadores encontra-se dividido em duas partes. Uma primeira parte que visa traçar o perfil do utilizador de modo contextualizar a sua participação e outra parte destinada à sua opinião pessoal relativamente à usabilidade da aplicação e à sua satisfação com a experiência oferecida. A versão completa deste questionário, assim como os resultados obtidos neste inquérito, podem ser consultados detalhadamente no anexo A e no anexo B, respectivamente.

A análise dos resultados obtidos revela que a maioria dos inquiridos aprendeu rapidamente a utilizar a aplicação e considerou que a aplicação é de fácil utilização. Globalmente, os inquiridos manifestaram o seu agrado relativamente ao desenho da interface e ao meio de interacção usado, afirmando que a interface respondeu rapidamente aos seus movimentos e permitiu recuperar facilmente de situações inesperadas. Destaque ainda para o facto de todos os inquiridos terem afirmado que gostaram de utilizar a aplicação, reforçando a ideia que a aplicação foi bem recebida junto do potencial público.

Contudo, apesar da recepção positiva, foram identificadas algumas dificuldades por parte dos utilizadores, expressas verbalmente durante a utilização, assim como textualmente numa

secção dedicada a comentários e sugestões presente no fim do questionário, que merecem destaque.

Apesar da maioria dos inquiridos afirmar não ter encontrado quaisquer inconsistências na utilização da aplicação, com alguns inquiridos, inclusive, a deixar comentários elogiosos relativos à clareza e usabilidade da interface, alguns utilizadores identificaram determinadas inconsistências ao nível do desenho da interface, relacionadas, sobretudo, com o *design* dos botões da aplicação. Alguns comentários referem que determinadas janelas da interface incluem imagens circulares com uma aparência similar à dos botões usados na aplicação, acabando por induzir o utilizador em erro, visto que as imagens não apresentam as mesmas funcionalidades. Como solução, foi sugerida a alteração do *design* dessas imagens ou a procura de uma solução subtil para distinguir entre botões e imagens simples.

Para além do *design* dos botões, foi também referido que apesar de divertido e viciante movimentar os botões na interface, essa funcionalidade acaba por não ter qualquer consequência prática. Alguns utilizadores referiram que essa funcionalidade deveria ser aproveitada para outras finalidades, com vista à adição de novas formas de navegação e interacção com a informação. Na sequência destes comentários, alguns utilizadores referiram também que determinadas janelas da interface são demasiado estáticas e pouco interactivas, tendo sido sugerida a inclusão de animações e novos mecanismos interactivos para explorar e consultar a informação.

Por fim, alguns utilizadores consideraram a exploração dos códices virtuais demorada e cansativa. Segundo os inquiridos, obrigar os utilizadores a recriar totalmente o movimento de folhear é demasiado limitativo e não incentiva à consulta e exploração dos códices. Como solução, foi sugerida a inclusão de botões que permitam folhear automaticamente com a mesma animação, dispensando a recriação do movimento por parte dos utilizadores e assim atenuando consideravelmente os problemas identificados. Foi ainda criticada a impossibilidade de navegar rapidamente para qualquer secção dos códices, tendo sido sugerida a adição de uma barra ou de um índice remissivo, que permitam navegar rapidamente na estrutura das obras, e ainda a introdução de um método que permita indicar explicitamente o número do fólio pretendido de forma a navegar rapidamente para essa posição.



Conclusões e trabalho futuro

Neste capítulo é feita uma análise do trabalho realizado no âmbito desta dissertação e é apresentado o trabalho a desenvolver futuramente para a evolução da instalação interactiva.

6.1 Conclusões

O objectivo desta dissertação foi desenvolver uma instalação interactiva capaz de divulgar os resultados de um estudo sobre a cor em iluminuras medievais Portuguesas, assim como o tema das iluminuras em geral, através da exploração de tecnologias de interacção computacional, com a integração do real e do virtual, de uma forma envolvente, lúdica e adequada a espaços de interesse cultural, como museus e bibliotecas.

A instalação interactiva integra três aplicações distintas. Uma aplicação desenvolvida para Tablet PCs, que oferece uma experiência individual de produção de iluminuras, uma aplicação desenvolvida para uma superfície interactiva multitoque de grandes dimensões, que reúne num só local toda a informação disponível sobre o tema, e uma aplicação com uma interface tangível para a exploração de códigos virtuais. No âmbito desta dissertação foram desenvolvidas as duas primeiras aplicações, assim como um protótipo para avaliar a viabilidade da terceira aplicação.

As duas aplicações desenvolvidas assentam em dois paradigmas de interacção consideravelmente diferentes, nomeadamente, interfaces caligráficas e interfaces multitoque. Foi pelo estudo destes dois paradigmas de interacção, assim como pela sua aplicação à realidade dos manuscritos iluminados, que começou o trabalho desta dissertação. Foram analisadas as características subjacentes a cada um destes paradigmas de interacção e foram estudados vários exemplos práticos de sistemas pensados de raiz para estes paradigmas.

A realização da instalação interactiva dividiu-se em duas partes, primeiro foi desenvolvida

a aplicação Scriptorium Virtual e depois o painel interactivo para exploração de iluminuras medievais.

A aplicação Scriptorium Virtual foi desenvolvida em Qt, uma plataforma que oferece suporte à construção de aplicações com interfaces caligráficas. Foram desenvolvidas funcionalidades associadas ao carregamento da informação semântica, ao desenho e pintura de iluminuras, à preparação de tintas e à consulta e exploração de informação.

O painel interactivo com interface multitoque, por sua vez, foi desenvolvido em openFrameworks, uma biblioteca que oferece as condições ideais para uma aplicação destas características. Foram desenvolvidos componentes para a detecção de eventos multitoque numa superfície interactiva de grandes dimensões, para o carregamento e apresentação de vários tipos de informação e ainda várias formas de interagir e explorar essa mesma informação.

A avaliação do trabalho realizado foi feita faseadamente, à medida que as aplicações foram desenvolvidas.

A avaliação da aplicação Scriptorium Virtual aconteceu em dois momentos distintos. Primeiro foi feita uma avaliação dedicada a peritos do Departamento de Conservação e Restauro da FCT/UNL, que consistiu na observação das suas interações e registo de comentários e sugestões, e mais tarde foi feita uma avaliação numa situação de utilização real, junto do seu público-alvo, onde foram feitos inquéritos aos utilizadores para avaliar a sua satisfação relativamente à aplicação utilizada.

A avaliação do painel interactivo, por sua vez, consistiu num inquérito aos utilizadores, após explorarem o sistema, de modo a aferir a eficácia dos meios de interacção usados e do desenho da interface.

Os resultados, em geral, foram bastante positivos para ambas as aplicações. As avaliações permitiram não só concluir que os objectivos propostos inicialmente foram atingidos, como identificar alguns aspectos a melhorar futuramente, com vista à evolução da instalação interactiva.

6.2 Trabalho futuro

O trabalho a desenvolver consiste, sobretudo, no aperfeiçoamento das aplicações desenvolvidas e na integração de novas funcionalidades, introduzindo maior diversidade e integrando mais conteúdos, sempre com o objectivo de melhorar a experiência oferecida aos utilizadores.

Relativamente à aplicação Scriptorium Virtual, os novos desenvolvimentos dizem respeito, sobretudo, a uma das principais dificuldades encontradas nas avaliações, nomeadamente, a dificuldade em reproduzir determinados detalhes das iluminuras. Apesar de existir a possibilidade de minimizar esta situação quando a aplicação for implementada em modelos de tablets mais recentes, a criação de um mecanismo de *zoom* eliminaria completamente quaisquer dificuldades, pois permitiria aos utilizadores aumentar as áreas mais problemáticas e desenhar e pintar com relativa facilidade. Para além disso, existe ainda a possibilidade de aperfeiçoar os traços desenhados pelos utilizadores, ajustando a sensibilidade à pressão e optimizando os parâmetros associados a cada um dos instrumentos de pintura utilizados.

O trabalho a desenvolver futuramente no painel interactivo consiste, essencialmente, na melhoria de algumas funcionalidades apresentadas, como a exploração de códices virtuais, e na adição de novas formas de interagir e visualizar os conteúdos, como a integração de gestos multitoque para manipular informação, a adaptação de algumas janelas a um conceito de participação colaborativa com vários utilizadores em simultâneo e, globalmente, a introdução de mais animações nas interacções de forma a imprimir um maior dinamismo à interface da aplicação. Em relação aos códices virtuais, algumas melhorias que podem ser implementadas dizem respeito à introdução de capas rígidas e à adição de botões, índices remissivos e barras de navegação para facilitar a sua exploração.

Os restantes aspectos encontram-se relacionados com o aperfeiçoamento do desenho das interfaces, sobretudo algumas particularidades identificadas nas avaliações, e com a integração de mais conteúdos nas duas aplicações.

Graças à natureza particionada da instalação interactiva existem vários caminhos possíveis para a sua evolução, quer seja através da introdução de novas aplicações, quer seja através do aperfeiçoamento e extensão das aplicações desenvolvidas no âmbito desta dissertação.

Bibliografia

- [AG04] Georg Apitz e François Guimbretière. Crossy: a crossing-based drawing application. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pág. 3–12, New York, NY, USA, 2004. ACM.
- [BGC10a] Daniele Borghesani, Costantino Grana, e Rita Cucchiara. Rerum novarum: interactive exploration of illuminated manuscripts. In *Proceedings of the international conference on Multimedia*, MM '10, pág. 1621–1624, New York, NY, USA, 2010. ACM.
- [BGC10b] Daniele Borghesani, Costantino Grana, e Rita Cucchiara. Surfing on artistic documents with visually assisted tagging. In *Proceedings of the international conference on Multimedia*, MM '10, pág. 1343–1352, New York, NY, USA, 2010. ACM.
- [Bha11] Sham Bhangal. The page turn effect in flash mx. <http://oreilly.com/javascript/archive/flashhacks.html>, Setembro 2011.
- [BSX⁺10] Leonardo Bonanni, Maurizio Seracini, Xiao Xiao, Matthew Hockenberry, Bianca Costanzo, Andrew Shum, Romain Teil, Antony Speranza, e Hiroshi Ishii. Tangible interfaces for art restoration. In *International Journal of Creative Interfaces and Computer Graphics*, pág. 54–66. IGI Global, 2010.
- [CMN⁺10] Nuno Correia, Tarquínio Mota, Rui Nóbrega, Luís Silva, e Andreia Almeida. A multi-touch tabletop for robust multimedia interaction in museums. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces 2010*, New York, NY, USA, 2010. ACM.
- [D.11] Wallin D. Touchlib: an opensource multi-touch framework (2006). <http://www.whitenoiseaudio.com/touchlib/>, Setembro 2011.
- [FIB95] George W. Fitzmaurice, Hiroshi Ishii, e William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pág. 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

- [Gro11] NUI Group. <http://www.nuigroup.com/>, Setembro 2011.
- [Han05] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pág. 115–118, New York, NY, USA, 2005. ACM.
- [HSCJ09] Michael S. Horn, Erin Treacy Solovey, R. Jordan Crouser, e Robert J.K. Jacob. Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pág. 975–984, New York, NY, USA, 2009. ACM.
- [HSJ08] Michael S. Horn, Erin Treacy Solovey, e Robert J. K. Jacob. Tangible programming and informal science learning: making tuis work for museums. In *Proceedings of the 7th international conference on Interaction design and children*, IDC '08, pág. 194–201, New York, NY, USA, 2008. ACM.
- [Hun09] Seth E. Hunter. Memtable - contextual memory in group workspaces. Tese de Mestrado, Massachusetts Institute of Technology, September, 2009.
- [IC01] Takeo Igarashi e Dennis Cosgrove. Adaptive unwrapping for interactive texture painting. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, I3D '01, pág. 209–216, New York, NY, USA, 2001. ACM.
- [IH03] Takeo Igarashi e John F. Hughes. Smooth meshes for sketch-based freeform modeling. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, I3D '03, pág. 139–142, New York, NY, USA, 2003. ACM.
- [IMT99] Takeo Igarashi, Satoshi Matsuoka, e Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pág. 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [IU97] Hiroshi Ishii e Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pág. 234–241, New York, NY, USA, 1997. ACM.
- [JGAK07] Sergi Jordà, Günter Geiger, Marcos Alonso, e Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pág. 139–146, New York, NY, USA, 2007. ACM.
- [KB07] Martin Kaltenbrunner e Ross Bencina. reactivation: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pág. 69–74, New York, NY, USA, 2007. ACM.

- [KBBC05] Martin Kaltenbrunner, Till Bovermann, Ross Bencina, e Enrico Costanza. TUIO - A Protocol for Table Based Tangible User Interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005.
- [LaL94] Mary LaLomia. User acceptance of handwritten recognition accuracy. In *Conference companion on Human factors in computing systems, CHI '94*, pág. 107–108, New York, NY, USA, 1994. ACM.
- [LW11] Zachary Lieberman e Theo Watson. openframeworks. <http://www.openframeworks.cc/>, Setembro 2011.
- [Mey95] André Meyer. Pen computing: a technology overview and a vision. *SIGCHI Bull.*, 27:46–90, July 1995.
- [Ope11] OpenCV. <http://opencv.willowgarage.com/wiki/>, Setembro 2011.
- [TMO11] The making of a medieval manuscript. http://www.fitzmuseum.cam.ac.uk/pharos/images/swf/manuscript/manuscript_5a.html, Setembro 2011.
- [Wei91] Mark Weiser. The computer for the 21st century. *Scientific American* 265, pág. 94–104, September 1991.
- [WFM03] Matthew Wright, Adrian Freed, e Ali Momeni. Opensound control: state of the art 2003. In *Proceedings of the 2003 conference on New interfaces for musical expression, NIME '03*, pág. 153–160, Singapore, Singapore, 2003. National University of Singapore.
- [z111] Community core vision. <http://ccv.nuigroup.com/>, Setembro 2011.
- [z211] Microsoft surface 2.0. <http://www.microsoft.com/surface/>, Setembro 2011.
- [z311] Rerum novarum. http://imagelab.ing.unimore.it/rerumnovarum/RerumNovarumHome/Rerum_Novarum.html, Setembro 2011.
- [z411] Turning the pagesTM. <http://www.turningthepages.com/>, Setembro 2011.
- [z511] Qt. <http://qt.nokia.com/>, Setembro 2011.



Questionários

Neste anexo são apresentados os questionários utilizados na avaliação das duas aplicações desenvolvidas. Primeiro é apresentado o questionário utilizado na avaliação da aplicação Scriptorium Virtual e depois o questionário utilizado na avaliação do painel interactivo para exploração de iluminuras medievais.

Questionário

Este questionário tem como objectivo a avaliação da interacção dos utilizadores com a aplicação Scriptorium Virtual, a correr num Tablet PC.

Todos os dados recolhidos são confidenciais e não serão utilizados para qualquer outra finalidade.

Utilizador

1. Idade

Idade: _____

2. Sexo

☐ Masculino ☐ Feminino

3. Grau de instrução

- | | |
|--|--|
| <input type="checkbox"/> Doutoramento/Pós-Doutoramento | <input type="checkbox"/> Mestrado/Pós-Graduação |
| <input type="checkbox"/> Licenciatura (completa) | <input type="checkbox"/> Licenciatura (incompleta) |
| <input type="checkbox"/> Ensino Secundário | <input type="checkbox"/> 3º Ciclo do Ensino Básico |
| <input type="checkbox"/> 2º Ciclo do Ensino Básico | <input type="checkbox"/> 1º Ciclo do Ensino Básico |
| <input type="checkbox"/> Autodidacta | |

4. Novas Tecnologias

4.1. Está familiarizado com as novas tecnologias? ☐ Sim ☐ Não

4.2. Que tipo de tecnologia utiliza?

	Diariamente	Semanalmente	Mensalmente	Raramente
Computador	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Telemóvel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consolas de jogos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Internet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Outra: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Scriptorium Virtual

Responda às seguintes questões fazendo um círculo em volta do número que melhor representa a sua opinião em relação à aplicação que acaba de experimentar.

5. Aprendizagem e Usabilidade

	Discordo			Concordo	
5.1. É fácil aprender a usar a aplicação.	1	2	3	4	5
5.2. Fiquei rapidamente habilitado a usar a aplicação.	1	2	3	4	5
5.3. Lembro-me facilmente de como usar a aplicação.	1	2	3	4	5
5.4. Consigo usar a aplicação sem necessitar de instruções escritas.	1	2	3	4	5
5.5. A aplicação é fácil de utilizar.	1	2	3	4	5
5.6. A interacção com caneta facilita a utilização da aplicação.	1	2	3	4	5
5.7. A aplicação responde rapidamente às diversas acções do utilizador.	1	2	3	4	5
5.8. Foi fácil recuperar de uma situação inesperada.	1	2	3	4	5
5.9. Não detectei inconsistências de utilização na interface.	1	2	3	4	5
5.10. Gostei de utilizar a aplicação.	1	2	3	4	5

5.11. Classifique a facilidade de execução das seguintes operações:

	Difícil			Fácil		Não Executou
a) Interagir com a aplicação através da caneta.	1	2	3	4	5	<input type="checkbox"/>
b) Reproduzir fielmente a iluminura escolhida.	1	2	3	4	5	<input type="checkbox"/>
c) Acompanhar o tutorial de produção de iluminuras.	1	2	3	4	5	<input type="checkbox"/>
d) Seleccionar as tintas desejadas.	1	2	3	4	5	<input type="checkbox"/>
e) Desenhar e escrever o pretendido.	1	2	3	4	5	<input type="checkbox"/>
f) Consultar informação sobre a iluminura.	1	2	3	4	5	<input type="checkbox"/>
g) Explorar a informação disponível.	1	2	3	4	5	<input type="checkbox"/>
h) Preparar tintas.	1	2	3	4	5	<input type="checkbox"/>
i) Acompanhar o tutorial na preparação das tintas.	1	2	3	4	5	<input type="checkbox"/>
j) Identificar os materiais na preparação das tintas.	1	2	3	4	5	<input type="checkbox"/>
k) Seleccionar os materiais na preparação das tintas.	1	2	3	4	5	<input type="checkbox"/>
l) Arrastar materiais para a área de preparação.	1	2	3	4	5	<input type="checkbox"/>
m) Moer os materiais na área de preparação.	1	2	3	4	5	<input type="checkbox"/>

6. Envolvimento Emocional

Escolha as expressões que melhor definem a sua experiência na utilização da aplicação.

- | | | |
|---------------------------------------|---|--|
| <input type="checkbox"/> Agradável | <input type="checkbox"/> Simples | <input type="checkbox"/> Cansativa |
| <input type="checkbox"/> Frustrante | <input type="checkbox"/> Divertida | <input type="checkbox"/> Imersiva |
| <input type="checkbox"/> Estimulante | <input type="checkbox"/> Aborrecida | <input type="checkbox"/> Irritante |
| <input type="checkbox"/> Complexa | <input type="checkbox"/> Inovadora | <input type="checkbox"/> Entusiasmante |
| <input type="checkbox"/> Confusa | <input type="checkbox"/> Impressionante | <input type="checkbox"/> Motivante |
| <input type="checkbox"/> Péssima | <input type="checkbox"/> Ótima | <input type="checkbox"/> Interessante |
| <input type="checkbox"/> Satisfatória | <input type="checkbox"/> Muito técnica | <input type="checkbox"/> Sem valor |
| <input type="checkbox"/> Útil | <input type="checkbox"/> Viciante | <input type="checkbox"/> Informativa |

7. Sugestões e Comentários

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Questionário

Este questionário tem como objectivo a avaliação da interacção dos utilizadores com o painel interactivo para exploração de iluminuras medievais.

Todos os dados recolhidos são confidenciais e não serão utilizados para qualquer outra finalidade.

Utilizador

1. Idade

Idade: _____

2. Sexo

☐ Masculino ☐ Feminino

3. Grau de instrução

- | | |
|--|--|
| <input type="checkbox"/> Doutoramento/Pós-Doutoramento | <input type="checkbox"/> Mestrado/Pós-Graduação |
| <input type="checkbox"/> Licenciatura (completa) | <input type="checkbox"/> Licenciatura (incompleta) |
| <input type="checkbox"/> Ensino Secundário | <input type="checkbox"/> 3º Ciclo do Ensino Básico |
| <input type="checkbox"/> 2º Ciclo do Ensino Básico | <input type="checkbox"/> 1º Ciclo do Ensino Básico |
| <input type="checkbox"/> Autodidacta | |

4. Novas Tecnologias

4.1. Está familiarizado com as novas tecnologias? ☐ Sim ☐ Não

4.2. Que tipo de tecnologia utiliza?

	Diariamente	Semanalmente	Mensalmente	Raramente
Computador	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Telemóvel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consolas de jogos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Internet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Outra: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Painel interactivo para exploração das iluminuras medievais

Responda às seguintes questões fazendo um círculo em volta do número que melhor representa a sua opinião em relação à aplicação que acaba de experimentar.

5. Aprendizagem e Usabilidade

	Discordo			Concordo	
5.1. Foi fácil aprender a utilizar a aplicação.	1	2	3	4	5
5.2. A aplicação é fácil de utilizar.	1	2	3	4	5
5.3. A superfície de interacção facilita a navegação na interface.	1	2	3	4	5
5.4. A interface respondeu rapidamente aos meus movimentos.	1	2	3	4	5
5.5. Foi fácil recuperar de uma situação inesperada.	1	2	3	4	5
5.6. Não detectei inconsistências de utilização na interface.	1	2	3	4	5
5.7. Gostei de utilizar o painel interactivo.	1	2	3	4	5

5.11. Classifique a facilidade de execução das seguintes operações:

	Difícil			Fácil		Não Executou
a) Consultar e explorar informação.	1	2	3	4	5	<input type="checkbox"/>
b) Navegar na interface.	1	2	3	4	5	<input type="checkbox"/>
c) Navegar na galeria de iluminuras.	1	2	3	4	5	<input type="checkbox"/>
d) Explorar os códigos virtuais.	1	2	3	4	5	<input type="checkbox"/>

6. Envolvimento Emocional

Escolha as expressões que melhor definem a sua experiência na utilização da aplicação.

<input type="checkbox"/> Agradável	<input type="checkbox"/> Simples	<input type="checkbox"/> Cansativa
<input type="checkbox"/> Frustrante	<input type="checkbox"/> Divertida	<input type="checkbox"/> Imersiva
<input type="checkbox"/> Estimulante	<input type="checkbox"/> Aborrecida	<input type="checkbox"/> Irritante
<input type="checkbox"/> Complexa	<input type="checkbox"/> Inovadora	<input type="checkbox"/> Entusiasmante
<input type="checkbox"/> Confusa	<input type="checkbox"/> Impressionante	<input type="checkbox"/> Motivante
<input type="checkbox"/> Péssima	<input type="checkbox"/> Óptima	<input type="checkbox"/> Interessante
<input type="checkbox"/> Satisfatória	<input type="checkbox"/> Muito técnica	<input type="checkbox"/> Sem valor
<input type="checkbox"/> Útil	<input type="checkbox"/> Viciante	<input type="checkbox"/> Informativa

7. Sugestões e Comentários

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



Resultados dos inquéritos

Neste anexo são apresentados os resultados obtidos nos inquéritos realizados no âmbito da avaliação das aplicações desenvolvidas. Primeiro são apresentados os resultados obtidos no inquérito relativo à aplicação Scriptorium Virtual e depois os resultados obtidos no inquérito respeitante ao painel interactivo para exploração de iluminuras medievais.

B.1 Scriptorium Virtual

Faixa etária	Participantes
Menos de 10	0
Entre 10 e 19	10
Entre 20 e 29	3
Entre 30 e 39	1
Entre 40 e 49	0
Entre 50 e 59	0
Entre 60 e 69	0
Mais de 70	0

Sexo	Participantes
Feminino	7
Masculino	7

Grau de Instrução	Participantes
Doutoramento/Pós-Doutoramento	0
Mestrado/Pós-Graduação	1
Licenciatura (completa)	1
Licenciatura (incompleta)	2
Ensino Secundário	8
3º Ciclo do Ensino Básico	2
2º Ciclo do Ensino Básico	0
1º Ciclo de Ensino Básico	0
Autodidacta	0

Está familiarizado com as novas tecnologias?	Participantes
Sim	14
Não	0

Que tipo de tecnologia utiliza?	Diariamente	Semanalmente	Mensalmente	Raramente
Computador	14	0	0	0
Telemóvel	14	0	0	0
Consolas de jogos	0	3	2	9
Internet	14	0	0	0

Aprendizagem e usabilidade	1	2	3	4	5
É fácil aprender a usar a aplicação.	0	0	1	7	6
Fiquei rapidamente habilitado a usar a aplicação.	0	0	2	5	7
Lembro-me facilmente de como usar a aplicação.	0	0	0	7	7
Consigo usar a aplicação sem necessitar de instruções escritas.	2	1	5	3	3
A aplicação é fácil de utilizar.	0	0	2	9	3
A interacção com caneta facilita a utilização da aplicação.	0	0	3	6	5
A aplicação responde rapidamente às diversas acções do utilizador.	0	0	4	3	7
Foi fácil recuperar de uma situação inesperada.	0	2	3	5	4
Não detectei inconsistências de utilização na interface.	0	0	3	4	7
Gostei de utilizar a aplicação.	0	0	0	2	12

Aprendizagem e usabilidade	1	2	3	4	5	Não executou
Interagir com a aplicação através da caneta.	0	1	0	6	7	0
Reproduzir fielmente a iluminura escolhida.	1	2	4	6	1	0
Acompanhar o tutorial de produção de iluminuras.	0	0	3	5	6	0
Seleccionar as tintas desejadas.	0	0	0	4	10	0
Desenhar e escrever o pretendido.	0	1	3	9	1	0
Consultar informação sobre a iluminura.	0	0	3	4	7	0
Explorar a informação disponível.	0	0	4	4	5	1
Preparar tintas.	0	0	2	4	8	0
Acompanhar o tutorial na preparação das tintas.	0	0	2	5	7	0
Identificar os materiais na preparação das tintas.	0	0	3	4	7	0
Seleccionar os materiais na preparação das tintas.	0	0	2	5	7	0
Arrastar materiais para a área de preparação.	0	0	2	4	8	0
Moer os materiais na área de preparação.	0	0	2	4	7	1

Envolvimento emocional	Participantes
Agradável	11
Frustrante	0
Estimulante	7
Complexa	2
Confusa	0
Péssima	0
Satisfatória	4
Útil	6
Simples	3
Divertida	10
Aborrecida	0
Inovadora	9
Impressionante	2
Ótima	3
Muito técnica	2
Viciante	1
Cansativa	0
Imersiva	2
Irritante	0
Entusiasmante	5
Motivante	6
Interessante	10
Sem valor	0
Informativa	8

B.2 Painel interactivo para exploração de iluminuras medievais

Faixa etária	Participantes
Menos de 10	0
Entre 10 e 19	1
Entre 20 e 29	5
Entre 30 e 39	2
Entre 40 e 49	1
Entre 50 e 59	0
Entre 60 e 69	0
Mais de 70	0

Sexo	Participantes
Feminino	6
Masculino	3

Grau de Instrução	Participantes
Doutoramento/Pós-Doutoramento	0
Mestrado/Pós-Graduação	3
Licenciatura (completa)	2
Licenciatura (incompleta)	1
Ensino Secundário	3
3º Ciclo do Ensino Básico	0
2º Ciclo do Ensino Básico	0
1º Ciclo de Ensino Básico	0
Autodidacta	0

Está familiarizado com as novas tecnologias?	Participantes
Sim	9
Não	0

Que tipo de tecnologia utiliza?	Diariamente	Semanalmente	Mensalmente	Raramente
Computador	8	1	0	0
Telemóvel	9	0	0	0
Consolas de jogos	0	2	3	4
Internet	9	0	0	0

Aprendizagem e usabilidade	1	2	3	4	5
Foi fácil aprender a utilizar a aplicação.	0	0	0	3	6
A aplicação é fácil de utilizar.	0	0	0	1	8
A superfície de interacção facilita a navegação na interface.	0	0	4	1	4
A interface respondeu rapidamente aos meus movimentos.	0	0	0	1	8
Foi fácil recuperar de uma situação inesperada.	0	0	0	1	8
Não detectei inconsistências de utilização na interface.	0	0	0	1	8
Gostei de utilizar o painel interactivo.	0	0	0	0	9

Aprendizagem e usabilidade	1	2	3	4	5	Não executou
Consultar e explorar informação.	0	0	0	2	7	0
Navegar na interface.	0	0	0	0	9	0
Navegar na galeria de iluminuras.	0	0	0	1	6	2
Explorar os códigos virtuais.	0	0	2	2	4	1

Envolvimento emocional	Participantes
Agradável	9
Frustrante	0
Estimulante	0
Complexa	0
Confusa	0
Péssima	0
Satisfatória	3
Útil	3
Simples	9
Divertida	0
Aborrecida	1
Inovadora	0
Impressionante	0
Óptima	0
Muito técnica	0
Viciante	0
Cansativa	0
Imersiva	1
Irritante	0
Entusiasmante	0
Motivante	1
Interessante	0
Sem valor	0
Informativa	7



Formulário de observação

Neste anexo é apresentado o formulário de observação utilizado na avaliação da aplicação Scriptorium Virtual.

FORMULÁRIO DE OBSERVAÇÃO

UTILIZADOR

1. Idade

< 10	<input type="checkbox"/>	40 - 49	<input type="checkbox"/>
10 - 19	<input type="checkbox"/>	50 - 59	<input type="checkbox"/>
20 - 29	<input type="checkbox"/>	60 - 69	<input type="checkbox"/>
30 - 39	<input type="checkbox"/>	>= 70	<input type="checkbox"/>

2. Sexo

Masculino	<input type="checkbox"/>	Feminino	<input type="checkbox"/>
-----------	--------------------------	----------	--------------------------

INTERACÇÃO

3. Interface

	Reduzido			Elevado	
Tempo de aprendizagem	1	2	3	4	5
Dificuldade de utilização	1	2	3	4	5
Dificuldade em navegar na interface	1	2	3	4	5

Principais obstáculos	Desenho	<input type="checkbox"/>	Hardware	<input type="checkbox"/>	Outros	<input type="checkbox"/>
Quais?						

Tempo total de utilização (em minutos):

< 10	<input type="checkbox"/>	31 - 40	<input type="checkbox"/>
11 - 20	<input type="checkbox"/>	41 - 60	<input type="checkbox"/>
21 - 30	<input type="checkbox"/>	> 60	<input type="checkbox"/>

Observações:

4. Produção de iluminuras

				Reduzido				Elevado
Tempo de aprendizagem				1	2	3	4	5
Dificuldade de utilização				1	2	3	4	5

Principais obstáculos Desenho ☐ Hardware ☐ Outros ☐

Quais?

	Reduzido			Elevado	
Dificuldade em acompanhar tutorial	1	2	3	4	5
Dificuldade em criar iluminura	1	2	3	4	5
Dificuldade na selecção de tintas	1	2	3	4	5
Adequação ao workshop real	1	2	3	4	5
Expressividade oferecida pelo hardware	1	2	3	4	5
Satisfação em relação à iluminura criada	1	2	3	4	5
Satisfação global	1	2	3	4	5

Tempo de utilização (em minutos):

< 5 ☐ 26 – 35 ☐

6 – 15 ☐ 36 – 45 ☐

16 – 25 ☐ > 45 ☐

Observações:

5. Preparação de tintas

O utilizador explorou este painel? Sim ☐ Não ☐

				Reduzido				Elevado
Tempo de aprendizagem				1	2	3	4	5
Dificuldade de utilização				1	2	3	4	5

Principais obstáculos Desenho ☐ Hardware ☐ Outros ☐

Quais?

	Reduzido			Elevado	
Dificuldade em acompanhar tutorial	1	2	3	4	5
Dificuldade em preparar tinta escolhida	1	2	3	4	5
Adequação ao workshop real	1	2	3	4	5
Satisfação em relação aos mecanismos de interação usados	1	2	3	4	5
Satisfação global	1	2	3	4	5

Tempo de utilização (em minutos):

< 1	<input type="checkbox"/>	11 – 15	<input type="checkbox"/>
1 – 5	<input type="checkbox"/>	16 – 20	<input type="checkbox"/>
6 – 10	<input type="checkbox"/>	> 20	<input type="checkbox"/>

Observações:

6. Selecção de Iluminuras

O utilizador explorou este painel? Sim ☐ Não ☐

			Reduzido		Elevado	
Tempo de aprendizagem		1	2	3	4	5
Dificuldade de utilização		1	2	3	4	5

Principais obstáculos Desenho ☐ Hardware ☐ Outros ☐

Quais?

Tempo de utilização (em minutos):

< 1	<input type="checkbox"/>	11 – 15	<input type="checkbox"/>
1 – 5	<input type="checkbox"/>	16 – 20	<input type="checkbox"/>
6 – 10	<input type="checkbox"/>	> 20	<input type="checkbox"/>

Observações:

7. Painel Informativo

O utilizador explorou este painel? Sim ☐ Não ☐

			Reduzido		Elevado	
Tempo de aprendizagem		1	2	3	4	5
Dificuldade de utilização		1	2	3	4	5

Principais obstáculos Desenho ☐ Hardware ☐ Outros ☐

Quais?

Tempo de utilização (em minutos):

< 1	<input type="checkbox"/>	11 - 15	<input type="checkbox"/>
1 - 5	<input type="checkbox"/>	16 - 20	<input type="checkbox"/>
6 - 10	<input type="checkbox"/>	> 20	<input type="checkbox"/>

Observações:



Resultados da observação

Neste anexo são apresentados os resultados da observação de utilizadores respeitante à aplicação Scriptorium Virtual.

Faixa etária	Utilizadores
Menos de 10	0
Entre 10 e 19	0
Entre 20 e 29	1
Entre 30 e 39	1
Entre 40 e 49	2
Entre 50 e 59	0
Entre 60 e 69	0
Mais de 70	0

Sexo	Utilizadores
Feminino	4
Masculino	0

Interface: aprendizagem e usabilidade	1	2	3	4	5
Tempo de aprendizagem.	0	3	1	0	0
Dificuldade de utilização.	0	1	2	1	0
Dificuldade em navegar na interface.	0	0	3	1	0

Interface: principais obstáculos	Utilizadores
Desenho da interface.	4
Hardware.	1

Interface: tempo de utilização	Utilizadores
< 10 min	1
11 - 20 min	2
21 - 30 min	1
31 - 40 min	0
41 - 60 min	0
> 60 min	0

Produção de iluminuras: aprendizagem e usabilidade	1	2	3	4	5
Tempo de aprendizagem.	0	4	0	0	0
Dificuldade de utilização.	0	3	1	0	0
Dificuldade em acompanhar tutorial.	1	2	1	0	0
Dificuldade em criar iluminura.	1	2	0	1	0
Dificuldade na selecção de tintas.	4	0	0	0	0
Adequação ao workshop real.	0	0	0	4	0
Expressividade oferecida pelo hardware.	0	1	2	1	0
Satisfação em relação à iluminura criada.	0	0	3	1	0
Satisfação global.	0	0	3	1	0

Produção de iluminuras: principais obstáculos	Utilizadores
Desenho da interface.	1
Hardware.	4

Produção de iluminuras: tempo de utilização	Utilizadores
< 5 min	1
6 - 15 min	2
16 - 25 min	1
26 - 35 min	0
36 - 45 min	0
> 45 min	0

Preparação de tintas: o utilizador explorou este painel?	Utilizadores
Sim	4
Não	0

Preparação de tintas: aprendizagem e usabilidade	1	2	3	4	5
Tempo de aprendizagem.	1	3	0	0	0
Dificuldade de utilização.	0	3	1	0	0
Dificuldade em acompanhar tutorial.	1	1	1	1	0
Dificuldade em preparar tinta escolhida.	0	2	2	0	0
Adequação ao workshop real.	0	0	0	4	0
Satisfação em relação aos mecanismos de interacção usados.	0	0	0	0	4
Satisfação global.	0	0	0	3	1

Preparação de tintas: principais obstáculos	Utilizadores
Desenho da interface.	3
Hardware.	0

Preparação de tintas: tempo de utilização	Utilizadores
< 1 min	0
1 - 5 min	2
6 - 10 min	2
11 - 15 min	0
16 - 20 min	0
> 20 min	0

Seleção de iluminuras: o utilizador explorou este painel?	Utilizadores
Sim	4
Não	0

Seleção de iluminuras: aprendizagem e usabilidade	1	2	3	4	5
Tempo de aprendizagem.	3	1	0	0	0
Dificuldade de utilização.	3	1	0	0	0

Seleção de iluminuras: principais obstáculos	Utilizadores
Desenho da interface.	0
Hardware.	0

Seleção de iluminuras: tempo de utilização	Utilizadores
< 1 min	4
1 - 5 min	0
6 - 10 min	0
11 - 15 min	0
16 - 20 min	0
> 20 min	0

Painel informativo: o utilizador explorou este painel?	Utilizadores
Sim	2
Não	2

Painel informativo: aprendizagem e usabilidade	1	2	3	4	5
Tempo de aprendizagem.	1	1	0	0	0
Dificuldade de utilização.	1	0	1	0	0

Painel informativo: principais obstáculos	Utilizadores
Desenho da interface.	1
Hardware.	0

Painel informativo: tempo de utilização	Utilizadores
< 1 min	1
1 - 5 min	1
6 - 10 min	0
11 - 15 min	0
16 - 20 min	0
> 20 min	0



XML Schemas

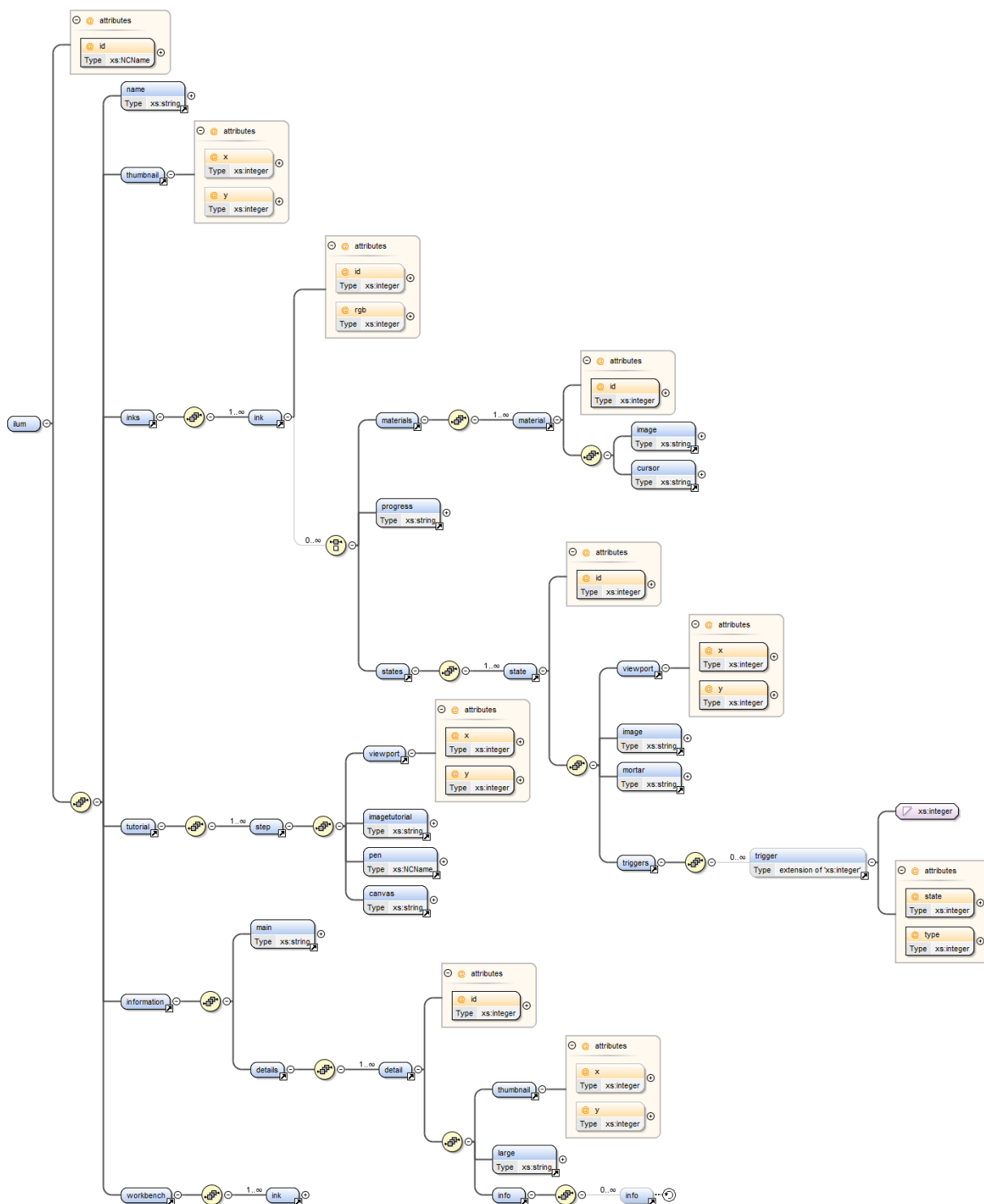


Figura E.1: XML Schema que define as regras de validação da aplicação Scriptorium Virtual

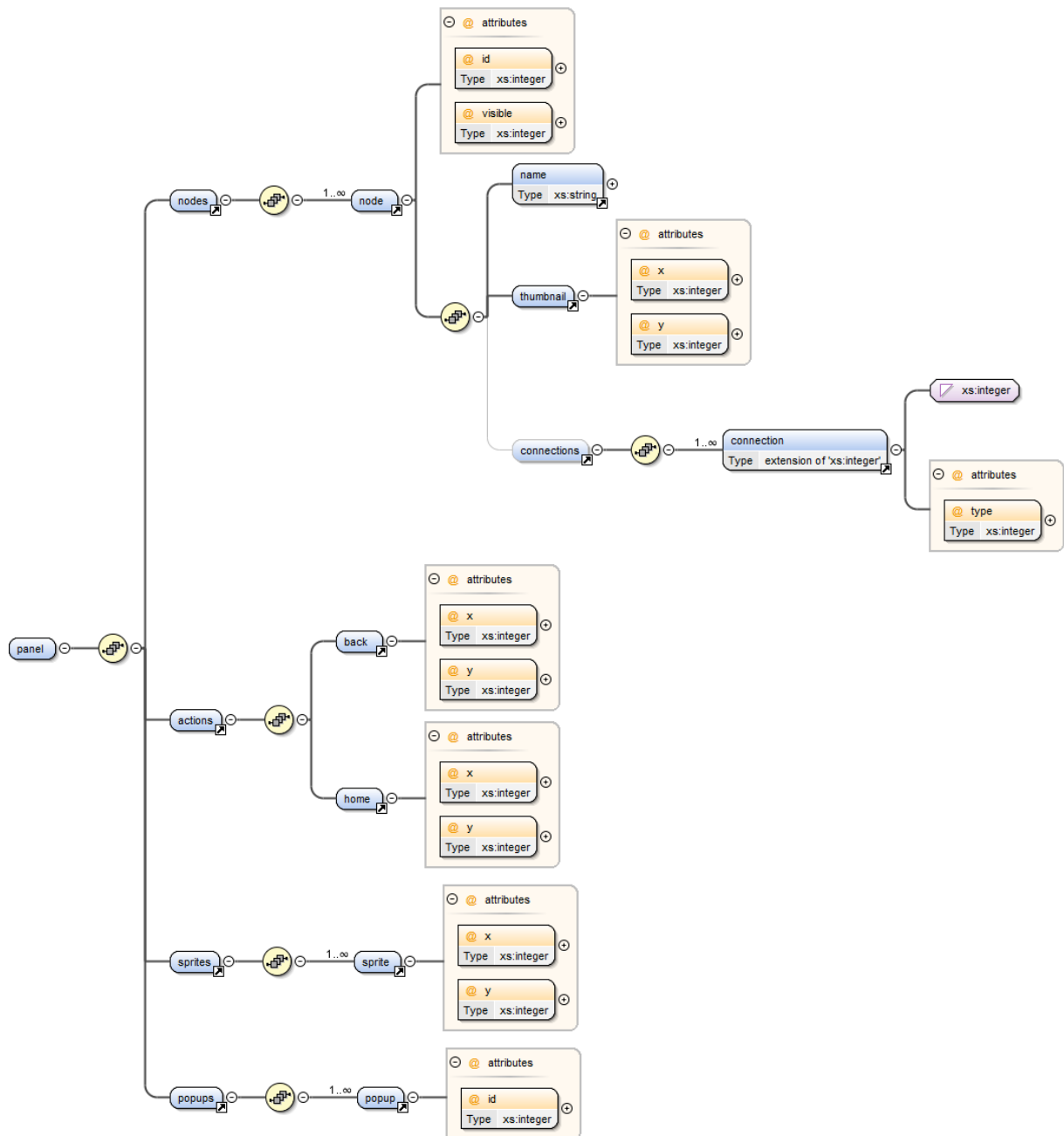


Figura E.2: XML Schema que define as regras de validação das janelas do painel interativo